M10 COMPUTER PORTATILE

Guida operativa

olivetti

PUBBLICAZIONE EMESSA DA:

Ing. C. Olivetti & C., S.p.A.
Direzione Documentazione
77, Via Jervis-10015 IVREA (Italy)

Copyright © 1983, by Olivetti
Tutti i diritti riservati

PREFAZIONE

Olivetti M10 è un piccolo computer portatile, tanto versatile da poter essere utilizzato con la massima facilità in ufficio, in casa, in viaggio o fuori sede.

Le caratteristiche più salienti di M10 sono la semplicità d'uso, un display a cristalli liquidi di otto righe, una tastiera di facile operabilità, la collegabilità di diverse periferiche, una gamma di funzioni di elaborazione testi, una potente programmabilità in linguaggio BASIC e la possibilità di collegamento locale o remoto con altri elaboratori.

Tutti questi aspetti di M10, ed altri ancora, sono descritti in questo manuale che, per comodità dell'utente, è diviso in due parti.

La parte I, la "Guida dell'utente", riguarda essenzialmente l'operatività su M10; essa fornisce una visione d'insieme del sistema e delle periferiche, descrive le prestazioni e le possibili applicazioni, spiegando in dettaglio come utilizzare le cinque funzioni integrate di cui M10 è corredato. In altri termini essa introduce l'utente a M10 e fornisce le necessarie spiegazioni su tutti gli aspetti del suo utilizzo.

La parte II è dedicata al BASIC, (Beginners All-purpose Symbolic Instruction Code), il linguaggio di programmazione ad alto livello di M10. E' intitolata "Linguaggio BASIC: Guida di riferimento" e va intesa come tale. Essa cioè non pretende di insegnare a programmare in BASIC, ma rappresenta piuttosto un riepilogo della struttura del linguaggio, della sintassi e dei comandi, cui l'utente può far riferimento nel caso in cui abbia incertezze sull'utilizzo del BASIC in ambiente M10. In particolare la guida comprende un elenco alfabetico di tutti i comandi, delle istruzioni e delle funzioni BASIC, con un sommario del loro scopo e formato d'uso, e con esempi di utilizzazione.

RIFERIMENTI: M10 Compendio Istruzioni

DISTRIBUZIONE: Generale (G)

PRIMA EDIZIONE: Dicembre 1983

INDICE

PAGINA	
1-1	1. INTRODUZIONE
1-1	NOTAZIONE
1-2	DESCRIZIONE GENERALE
1-3	DESCRIZIONE TECNICA
1-4	INTERFACCE E PERIFERICHE
1-5	TASTIERA E DISPLAY
1-10	CONNESSIONI DEL PANNELLO POSTERIORE
1-12	PANNELLO INFERIORE
2-1	2. ATTIVAZIONE DI M10
2-1	INSTALLAZIONE DELLE PILE
2-2	PRIMA ATTIVAZIONE
3-1	3. IL MENU PRINCIPALE
3-2	IMPOSTAZIONE DELL'ORA
3-2	REIMPOSTAZIONE DELL'ORA
3-3	REIMPOSTAZIONE DELLA DATA
3-3	REIMPOSTAZIONE DEL GIORNO
4-1	4. L'AMBIENTE BASIC
4-1	IL DISPLAY E LA MODALITA' OPERATIVE
4-2	STATO COMANDI
4-2	STATO ESECUZIONE
4-2	STATO EDITOR DI LINEA
4-3	TASTI FUNZIONE IN AMBIENTE BASIC

PAGINA

4-4	CARATTERI DI CONTROLLO IN BASIC
5-1	5. IL PROGRAMMA APPLICATIVO TEXT
5-1	NOMI DEI FILE NEL PROGRAMMA TEXT
5-3	SCRITTURA DI UN FILE TESTO
5-4	TASTI FUNZIONE E COMANDO IN TEXT
5-5	RICERCA DI UNA STRINGA DI CARATTERI IN TEXT
5-5	LA FUNZIONE 'LOAD' IN TEXT
5-6	LA FUNZIONE 'SAVE' IN TEXT
5-7	LA FUNZIONE 'SELECT'
5-9	UTILIZZO DELLA FUNZIONE 'COPY'
5-9	LA FUNZIONE 'CUT' E 'PASTE'
5-10	STAMPA DI UN FILE TEXT
5-10	LA FUNZIONE 'MENU'
5-10	SOMMARIO DEI TASTI FUNZIONE E COMANDO
5-11	CARATTERI DI CONTROLLO CORRISPODENTI
6-1	6. IL PROGRAMMA APPLICATIVO ADDRSS
6-1	CREAZIONE DEL FILE 'ADRS.DO'
6-2	UTILIZZO DEL PROGRAMMA ADDRSS
7-1	7. IL PROGRAMMA APPLICATIVO SCHEDL
7-1	PREPARAZIONE DEL FILE 'NOTE.DO'
7-2	UTILIZZO DEL PROGRAMMA SCHEDL
8-1	8. IL PROGRAMMA TELCOM

PAGINA

8-1 CONNESSIONE DI M10 A UNA LINEA TELEFONICA

8-2 MODALITA' ENTRY

8-2 I TASTI FUNZIONE IN MODALITA' ENTRY

8-3 MODALITA' TERMINAL

8-3 I TASTI FUNZIONE IN MODALITA' TERMINAL

8-4 I PARAMETRI DI COMUNICAZIONE DATI

8-6 SCAMBIO DATI CON UN SISTEMA OSPITE

8-6 ACCESSO ALLA MODALITA' TERMINAL

8-7 LOG-ON SU SISTEMA OSPITE

8-8 USO DELLE PRESTAZIONI DI INVIO E SCARICAMENTO FILE

9-1 9. UTILIZZO DEL REGISTRATORE DI NASTRO A CASSETTA

9-1 COLLEGAMENTO DI M10 A UN REGISTRATORE A CASSETTA

9-3 REGISTRAZIONE DI UN FILE SU NASTRO A CASSETTA

9-3 REGISTRAZIONE DI UN FILE TEXT

9-3 REGISTRAZIONE DI UN FILE BASIC

9-4 CARICAMENTO DI UN FILE DA NASTRO A CASSETTA

9-4 CARICAMENTO DI UN FILE IN MODALITA' TEXT

9-5 CARICAMENTO DI UN FILE BASIC

9-5 MANUTENZIONE

10-1 10. INTRODUZIONE AL BASIC

11-1 11. PROGRAMMI BASIC

11-1 ORGANIZZAZIONE

PAGINA

11-1	DOCUMENTAZIONE DI UN PROGRAMMA
11-1	REM
11-1	CREAZIONE DI UN PROGRAMMA
11-1	INTRODUZIONE DI UN PROGRAMMA
11-2	LISTA DI UN PROGRAMMA
11-2	REGISTRAZIONE DI UN PROGRAMMA
11-3	USO DI UN PROGRAMMA
11-3	CARICAMENTO DI UN PROGRAMMA
11-3	ESECUZIONE DI UN PROGRAMMA
11-4	STAMPA
11-4	CORREZIONE DEGLI ERRORI IN UN PROGRAMMA
11-4	ERRORI
11-4	RICERCA DEGLI ERRORI
11-4	MODIFICHE A UN PROGRAMMA
11-5	VARIAZIONI DI UN PROGRAMMA
11-5	'EDITING' DI UN PROGRAMMA
11-5	FUSIONE DI DUE PROGRAMMI
12-1	12. DATI E ARITMETICA
12-1	I DATI NEI PROGRAMMI BASIC
12-1	DATI STRINGA
12-1	DATI NUMERICI
12-2	CLASSIFICAZIONE DEI DATI

PAGINA

12-2 MODIFICA DELLA CLASSIFICAZIONE

12-3 CONVERSIONI NUMERICHE

12-4 MATRICI

12-5 ARITMETICA BASIC

12-5 OPERATORI DI CONFRONTO (RELAZIONALI)

12-6 OPERATORI NUMERICI

12-8 OPERATORI LOGICI

12-10 INTRODUZIONE DEI DATI

13-1 13. CARATTERISTICHE DI PROGRAMMAZIONE

13-1 SALTI

13-1 CICLI ITERATIVI (LOOP)

13-1 SOTTOPROGRAMMI

13-1 SOTTOPROGRAMMI IN LINGUAGGIO MACCHINA

14-1 14. FILE

14-1 APERTURA DI UN FILE

14-1 LETTURA DI UN FILE DATI

14-1 SCRITTURA DI UN FILE DATI

14-2 CHIUSURA DI UN FILE

15-1 15. ELENCO DEI COMANDI, FUNZIONI E ISTRUZIONI BASIC

15-1 ABS Funzione

15-1 ASC Funzione

15-2 ATN Funzione

PAGINA

15-2	BEEP Istruzione
15-2	CALL Istruzione
15-3	CDBL Funzione
15-3	CHR$ Funzione
15-4	CINT Funzione
15-4	CLEAR Comando
15-5	CLOAD, CLOAD?, CLOADM Comandi
15-5	CLOSE Istruzione
15-6	CLS Comando
15-6	COM ON/OFF/STOP Istruzioni
15-7	CONT Comando
15-7	COS Funzione
15-7	CSAVE, CSAVEM Comandi
15-8	CSNG Funzione
15-8	CSRLIN Funzione
15-8	DATA Istruzione
15-9	DATE$ Istruzione e Funzione
15-9	DAY$ Istruzione e Funzione
15-10	DIM Istruzione
15-10	EDIT Comando
15-11	END Istruzione
15-11	EOF Funzione
15-12	ERL/ERR Funzioni

PAGINA

15-12	ERROR Istruzione
15-13	EXP Funzione
15-13	FILES Comando
15-14	FIX Funzione
15-14	FOR...NEXT Istruzioni
15-15	FRE Funzione
15-16	GOSUB e RETURN Istruzioni
15-17	GOTO Istruzione
15-17	HIMEM Funzione
15-18	IF...GOTO/THEN...ELSE... Istruzioni
15-19	INKEY$ Funzione
15-19	INP Funzione
15-20	INPUT Instruzione
15-20	INPUT Istruzione
15-21	INPUT$ Funzione
15-21	INSTR Funzione
15-22	INT Funzione
15-23	IPL Comando
15-23	KEY Istruzione
15-24	KILL Comando
15-24	LCOPY Istruzione
15-25	LEFT$ Funzione

PAGINA

15-25 LEN Funzione

15-25 LET Istruzione

15-26 LINE Comando

15-26 LINE INPUT Istruzione

15-27 LINE INPUT Istruzione

15-27 LIST Comando

15-28 LLIST Comando

15-28 LOAD, LOADM, CLOAD, CLOADM Comandi

15-29 LOG funzione

15-30 LPOS Funzione

15-30 LPRINT, LPRINT USING Istruzioni

15-30 MAXFILES Funzione

15-31 MAXRAM Funzione

15-31 MENU Comando

15-31 MERGE Comando

15-32 MICROPLOTTER Comandi

15-38 MID$ Funzione

15-38 MOD Funzione

15-38 MOTOR ON/OFF Istruzioni

15-39 NAME Comando

15-39 NEW Comando

15-40 ON...GOSUB Istruzione

PAGINA

15-41	ON...GOTO Istruzione
15-42	OPEN Istruzione
15-43	OUT Istruzione
15-43	PEEK Funzione
15-43	POKE Istruzione
15-44	POS Funzione
15-44	POWER Istruzione
15-45	PRESET Istruzione
15-45	PRINT Istruzione
15-46	PRINT USING Istruzione
15-49	PRINT, PRINT USING (file) Istruzioni
15-50	PSET Istruzione
15-50	READ Istruzione
15-51	REM Istruzione
15-51	RESTORE Istruzione
15-51	RESUME Istruzione
15-52	RIGHT$ Funzione
15-52	RND Funzione
15-53	RUN, RUNM Comandi
15-53	SAVE, SAVEM Comandi
15-55	SCREEN Istruzione
15-55	SGN Funzione

PAGINA

15-55	SIN Funzione
15-56	SOUND Istruzione
15-57	SPACE$ Funzione
15-57	SQR Funzione
15-58	STOP Istruzione
15-58	STRING$ Funzione
15-59	STR$ Funzione
15-59	TAB Funzione
15-60	TAN Funzione
15-60	TIME$ Istruzione e funzione
15-61	VAL funzione
15-61	VARPTR Funzione
15-62	WIDTH Istruzionze
A-1	A. SOMMARIO DELLE SPECIFICHE TECNICHE
B-1	B. MESSAGGI D'ERRORE BASIC
C-1	C. TABELLE DI EQUIVALENZA DEI CODICI
C-1	CARATTERI DI CONTROLLO
C-4	CARATTERI GRAFICI
D-1	D. SEQUENZE DI ESCAPE IN BASIC E IN MODO TELECOM
E-1	E. PAROLE RISERVATE
F-1	F. IL MODEM COUPLER OLIVETTI MC10
F-2	CARATTERISTICHE TECNICHE

PAGINA

F-4 COMMUTATORI E INDICATORI LUMINOSI

F-6 INSTALLAZIONE

F-6 OPERAZIONI DI COMUNICAZIONE DATI CON MC10

PARTE I

GUIDA DELL'UTENTE

1. INTRODUZIONE

NOTAZIONE

Per maggiore chiarezza, in questo manuale sono state adottate in modo uniforme le seguenti convenzioni:

1. Le parole scritte a lettere maiuscole rappresentano comandi o istruzioni per il sistema e devono essere introdotte nel formato indicato, non necessariamente a lettere maiuscole, dal momento che M10 converte automaticamente ogni lettera in maiuscolo (fatta eccezione per la modalità TEXT).
   Ad esempio:

   PRINT TIME$ può essere introdotto come
   print time$ oppure
   PrinT tIMe$ ma non
   PRINT TIME $ modificando la spaziatura.

   Anche i simboli hanno una funzione precisa e devono essere introdotti così come sono descritti.

   Una lista di tutti i simboli utilizzati nel linguaggio di programmazione BASIC è riportata all'inizio della parte II.

2. Le parole scritte a lettere minuscole sono variabili che l'utente deve sostituire con i valori effettivi. Ad esempio, in

   PRINT filename

   PRINT rappresenta un'istruzione da introdurre così com'è, mentre l'utente deve sostituire il nome effettivo del file da stampare alla parola "filename".

3. L'elenco che segue definisce il significato di alcuni gruppi di parole e di simboli.

   A|B|C scegliere uno fra gli elementi indicati, A o B o C

   [ ] elementi (chiusi tra parentesi quadre) opzionali

   ... più valori possono essere inclusi in una stringa, ad es.

   DATA costante-1, costante-2, ..., costante-n

   <KEY> premere il tasto il cui nome è racchiuso tra le

parentesi angolari (simboli di minore e maggiore), ad es.

<ENTER> significa "premere ENTER"
<CTRL> + x significa "premere x mantenendo premuto anche il tasto CONTROL"
<GRPH + SHIFT> significa "premere GRPH e SHIFT insieme"

4. Quanto compare sul display è riportato nel manuale in grassetto, ad es.

File to edit?

DESCRIZIONE GENERALE

M10 è un personal computer di piccole dimensioni, di facile uso e veramente portatile. Esso è raffigurato in Figura 1.1.

Fig. 1-1 Il computer portatile M10

M10 è il computer ideale in numerose applicazioni, con le sue prestazioni di programmabilità, elaborazione di testi, telecomunicazione, gestione di indirizzario e di agenda elettronica.

Può funzionare sia autonomamente per mezzo di quattro pile alcaline a

secco da 1,5 V, sia collegato alla rete di alimentazione mediante adattatore. Pesa circa 1800 grammi, e misura 30 x 22 x 6 centimetri. Può quindi essere facilmente sistemato in una normale valigetta, il che lo rende strumento ideale per l'utilizzo in viaggio o fuori sede, così come in ufficio o in casa. M10 è fornito di un elegante contenitore morbido che lo protegge dagli urti e dalla polvere.

Caratteristicche notevoli di M10 sono la tastiera e il display. Le critiche più frequentemente rivolte ai computer portatili riguardano la difficoltà d'uso della tastiera e la scarsa leggibilità del display. Non è certo il caso di M10, che è invece dotato di una robusta tastiera di normali dimensioni e di un display a cristalli liquidi di 8 righe da 40 caratteri di grandi dimensioni.

DESCRIZIONE TECNICA

M10 è dotato di una memoria ROM (Read-Only Memory) di capacità 32K byte, in cui risiedono cinque funzioni integrate:

- interprete BASIC, per creare, memorizzare ed eseguire programmi;
- TEXT, per preparare ed elaborare testi;
- TELCOM, per comunicare dati a distanza (telecomunicazioni);
- ADDRSS, per allestire ed aggiornare una rubrica elettronica di indirizzi e numeri telefonici;
- SCHEDL, per gestire facilmente gli appuntamenti ed altre attività.

Le altre caratteristiche principali sono:

- 8K di memoria RAM (Random Access Memory), con possibilità di espansione sino a 32K a moduli di 8K, ed alimentazione separata con batterie ricaricabili al Ni-Cd.
- Microprocessore CMOS OKI 80C85, compatibile con INTEL 8085.
- Tastiera dotata di 188 caratteri alfanumerici e grafici; sono disponibili le tastiere dattilografiche standard italiana, francese, tedesca ed inglese.
- Tasti funzione addizionali per applicazioni speciali.
- Display inclinabile a cristalli liquidi (LCD), con regolazione manuale del contrasto.
- Interfacce per collegamento a unità periferiche.

M10 ha tre directory (tabelle contenenti i nomi e gli indirizzi dei documenti, o "file") in memoria. I nomi dei file vengono completati mediante un suffisso secondo la directory in cui sono allocati.

I file contraddistinti dal suffisso .BA sono programmi in linguaggio BASIC memorizzati in forma binaria.

I file con suffisso .CO sono programmi in linguaggio macchina.

I file con suffisso .DO sono quelli creati con il programma applicativo TEXT, o programmi BASIC memorizzati in formato ISO, o file dati BASIC (questo manuale tratta soltanto i file .DO creati sotto TEXT).

INTERFACCE E PERIFERICHE

M10 è provvisto di interfacce per il collegamento a diversi tipi di unità periferiche. La Figura 1-2 mostra una configurazione di M10 con periferiche collegate. Sono disponibili cavi per tutte le interfacce.

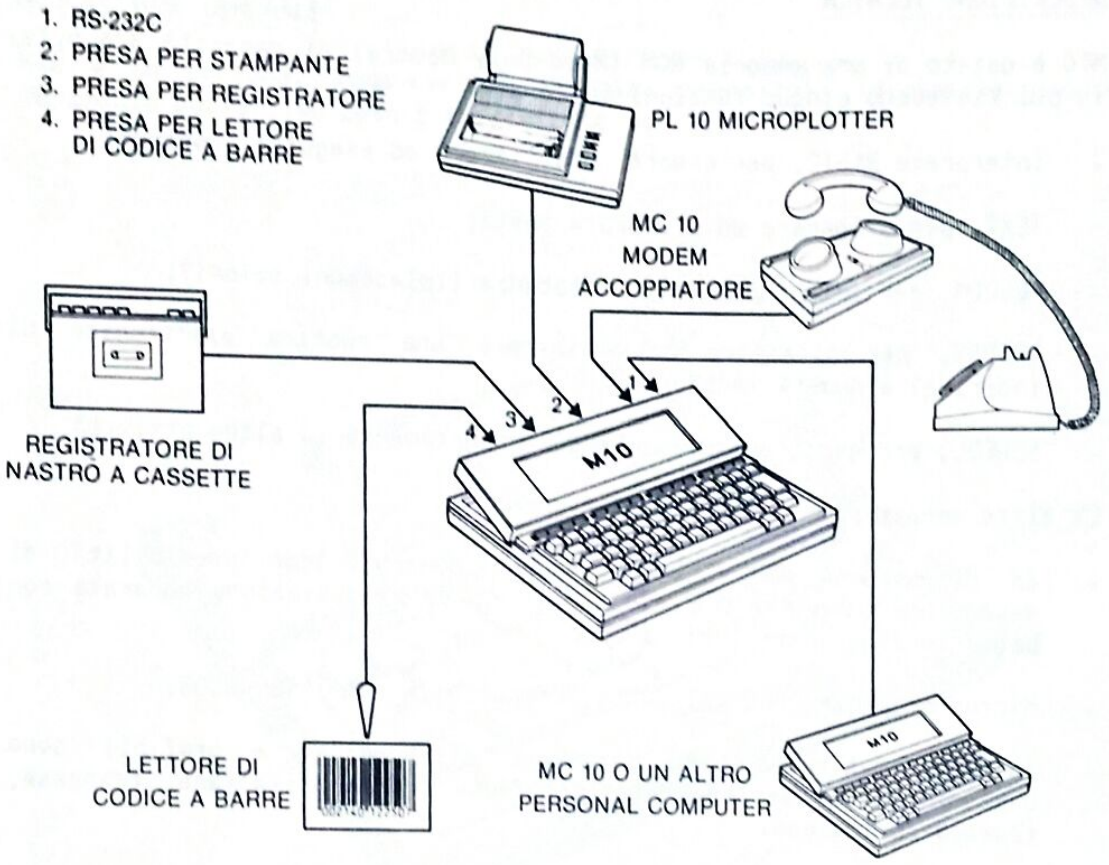

Fig 1-2 Configurazione di sistema.

Le principali periferiche sono:

Registratore di nastro a cassetta

M10 può collegarsi ai modelli di registratori di nastro a cassetta dotati, oltre che dei comandi standard, di jack di ingresso (MIC), di uscita (EAR), di controllo remoto (REM) e di contatore di nastro. Per ulteriori dettagli vedere il Capitolo 9.

Stampante

Stampanti o miniplotter dotati di interfaccia parallela conforme allo standard industriale possono essere collegati ad M10 tramite cavo miniplotter Olivetti.

Telefono

M10 può essere collegato ad elaboratori attraverso linea telefonica, utilizzando un modem/accoppiatore acustico come ad esempio il modello Olivetti MC10 (vedere Cap.8 e Appendice F per maggiori informazioni).

Lettore di codice a barre

M10 è provvisto di interfaccia per collegare un lettore di codice a barre, per applicazioni di gestione magazzini.

Tramite l'interfaccia RS-232C è inoltre possibile collegare M10 ad un altro elaboratore sia direttamente, con connessione locale per mezzo di un cavo "incrociato", che in collegamento remoto, tramite modem.

TASTIERA E DISPLAY

M10 è disponibile con tastiera dattilografica in versione nazionale italiana, francese, tedesca ed inglese (vedere ad es. la Figura 1-3). Tutti i modelli forniscono il set di caratteri alfanumerici completo.

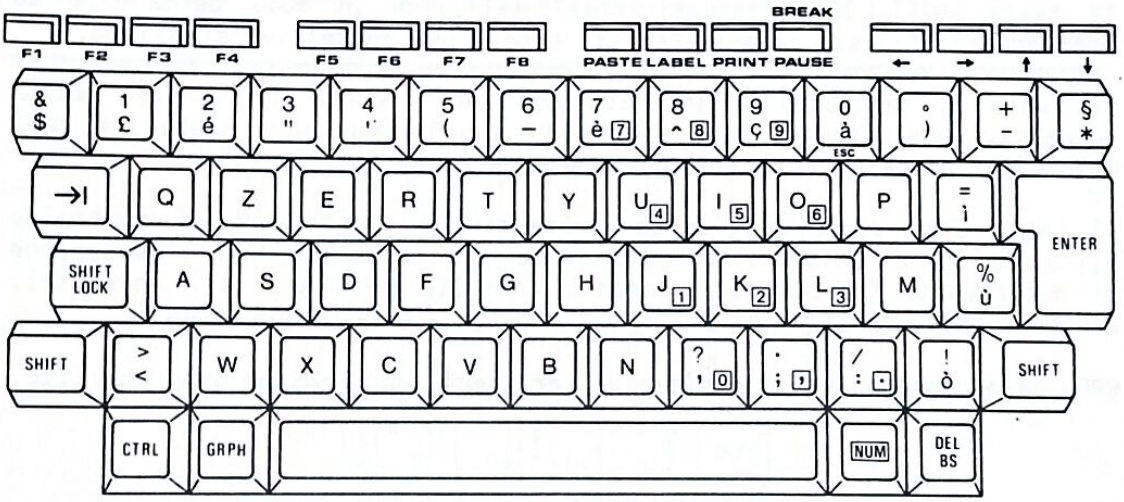

Fig. 1-3 Tastiera italiana

Premendo il tasto <GRPH>, sono accessibili altri 47 caratteri grafici, come è mostrato in Figura 1-4. Premendo invece <SHIFT + GRPH> si ottengono 47 ulteriori caratteri, come mostrato in Figura 1-5. Questi

insiemi di caratteri aggiuntivi sono identici in tutti i modelli.

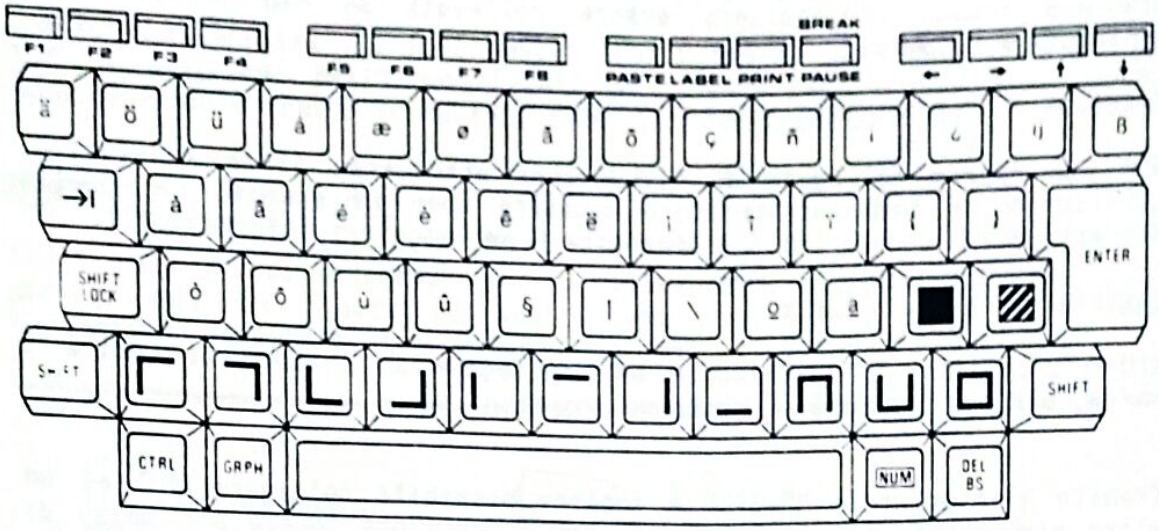

Fig. 1-4 Configurazione di tastiera quando viene premuto <GRPH>

Tutti i tasti carattere, come pure DEL/BS, il tasto di tabulazione (→|) e i tasti di movimento cursore, diventano ripetitivi se tenuti premuti.

Il tasto SHIFT (maiuscolo) svolge la stessa funzione che ricopre in una normale macchina per scrivere, ovvero seleziona il carattere maiuscolo nonchè i numeri e i simboli indicati nella parte superiore dei tasti.

Il tasto SHIFT LOCK (fissamaiuscole) seleziona in modo permanente la funzione SHIFT; esso rimane inserito sino a una successiva pressione.

Il tasto di tabulazione, indicato con →| , porta il cursore sulla successiva posizione di tabulazione. Le tabulazioni sono stabilite ogni otto spazi di carattere e non possono essere modificate.

Il tasto CTRL consente l'invio di particolari caratteri di controllo all'elaboratore. E' anche usato, insieme ad altri tasti, per l'esecuzione di determinate funzioni che saranno chiarite nei paragrafi specifici. L'elenco delle funzioni di controllo è riportato nell'Appendice C.

GRPH dà accesso ad altri insiemi di caratteri, che abbiamo già descritto.

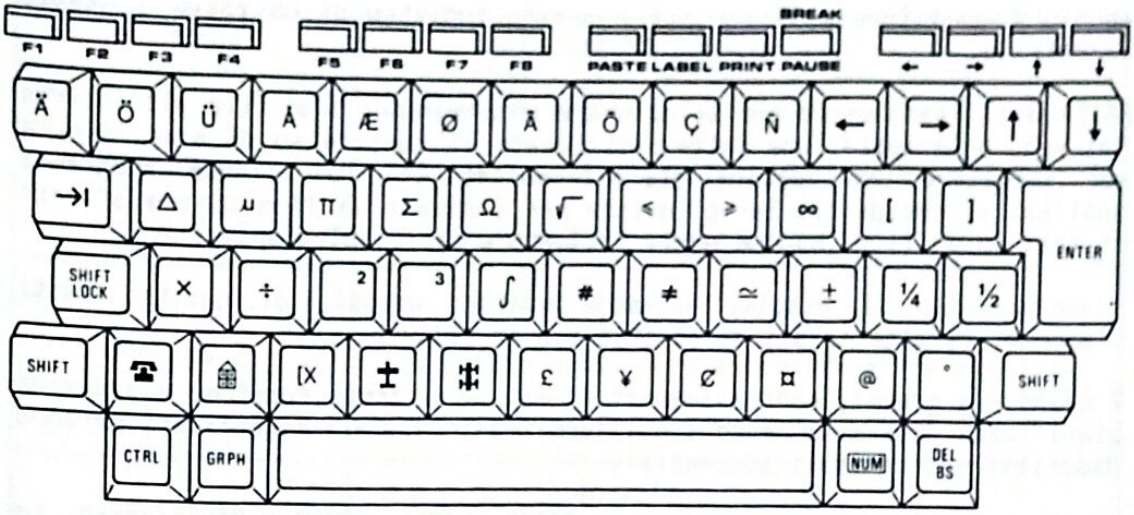

Fig. 1-5 Configurazione di tastiera quando viene premuto <GRPH + SHIFT>

Sul lato destro della tastiera vi sono 12 tasti che, oltre ai rispettivi caratteri normali, riportano le cifre da 0 a 9, la virgola e il punto, inscritti in un quadratino, come indicato nella Figura 1-6. Questa tastiera numerica (simile a quella delle macchine da calcolo) è attivata dalla pressione del tasto NUM, che rimane bloccato. In tale condizione tutti gli altri tasti sono inattivi; per tornare alla situazione precedente è sufficiente premere nuovamente il tasto NUM.

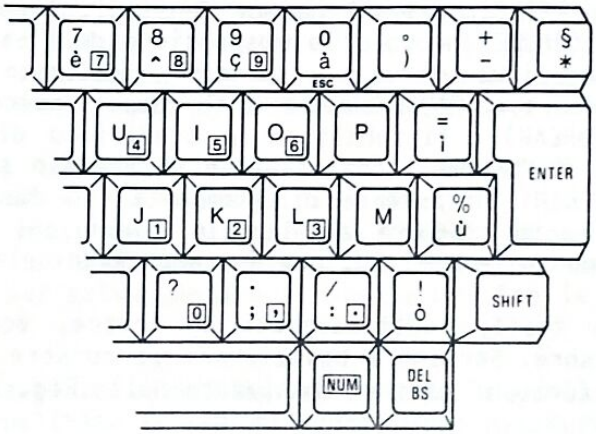

Fig. 1-6 Tastiera numerica, sovrapposta alla tastiera alfanumerica.

DEL/BS è il tasto di cancellazione/ritorno di un passo. Premuto da solo, cancella il carattere che precede il cursore, che arretra di uno spazio. Nella condizione TEXT, il comando <SHIFT + DEL/BS> cancella il carattere su cui è posizionato il cursore e sposta indietro di un passo il testo successivo.

ENTER è il tasto usato per selezionare un programma o un file all'interno del menù principale, una volta posizionato il cursore su di esso, oppure per scegliere una opzione all'interno di uno dei cinque programmi applicativi residenti. Serve inoltre per indicare la fine di una stringa di dati in un file, nonchè quale "ritorno a capo" sul display.

Proprio sotto il display vi sono quattro gruppi, di quattro tasti ciascuno, caratterizzati da scritte blu in rilievo.

I primi due gruppi, contrassegnati F1-F8, sono tasti funzione e assumono significati diversi a seconda dell'applicazione. Tali significati saranno descritti nei capitoli successivi.

Il terzo gruppo comprende i comandi PASTE, LABEL, PRINT e PAUSE/BREAK. Le funzioni di questi tasti sono sempre le stesse per tutti i programmi.

- PASTE (incolla), viene utilizzato nelle funzioni di 'Cut and Paste' (taglia e incolla) e di 'Copy' (copia). Queste sono tipiche funzioni di elaborazione testi e trovano quindi ovvia applicazione nel programma TEXT (descritto nel Capitolo 5).

- LABEL (etichetta), visualizza alla base del display il significato delle funzioni F1-F8 per il programma applicativo in uso. Premendo LABEL una seconda volta le informazioni scompaiono nuovamente.

- PRINT (stampa), viene utilizzato se M10 è collegato ad una stampante. Premuto da solo, provoca la stampa del contenuto del display. <SHIFT + PRINT> dà invece luogo alla stampa dell'intero file, permettendo inoltre di definire il numero di caratteri per riga di stampa.

- PAUSE/BREAK (pausa/interruzione) ha due funzioni. In ambiente BASIC, premuto da solo (PAUSE) provoca la sospensione dell'esecuzione di un programma; l'esecuzione viene ripresa quando il tasto viene premuto nuovamente. <SHIFT> + <PAUSE/BREAK> interrompe invece l'esecuzione del programma (BREAK) o ristabilisce la condizione di funzionamento normale in caso di "blocco", come avviene ad esempio se si esegue un un comando di PRINT in assenza di stampante. In fase di TEXT, la pressione del tasto <BREAK> annulla le operazioni di selezione, ricerca, caricamento, memorizzazione o stampa eventualmente in corso.

Gli ultimi quattro tasti, contrassegnati da frecce, sono i tasti di spostamento del cursore. Servono a posizionare il cursore nel display in fase TEXT; le loro funzioni sono schematizzate nella Figura 1-7.

TASTO	SPOSTAMENTO CURSORE
→	uno spazio a destra
<SHIFT> + →	all'inizio della parola seguente
<CTRL> + →	alla fine della riga in corso
←	uno spazio a sinistra
<SHIFT> + ←	all'inizio della parola precedente (o attuale)
<CTRL> + ←	all'inizio della riga in corso
↑	una riga più in alto
<SHIFT> + ↑	alla prima riga del display
<CTRL> + ↑	all'inizio del file
↓	una riga più in basso
<SHIFT> + ↓	all'ultima riga del display
<CTRL> + ↓	alla fine del file

Fig. 1-7 Tasti di spostamento cursore

Oltre alle funzioni descritte in tabella, i tasti di spostamento cursore hanno un effetto ripetitivo se mantenuti premuti. Tenendo premuta la freccia a destra, il cursore scorre da sinistra verso destra e da una riga all'altra verso il fondo del file. Con la freccia a sinistra si ha l'effetto contrario. Tenendo premuta la freccia in basso, il cursore si sposta verso il basso, riga per riga, nell'ambito del file (questa operazione viene chiamata "scroll", scorrimento); ovviamente si ottiene l'effetto contrario usando il tasto contrassegnato dalla freccia in alto.

Lo schermo, sul quale sono visualizzati i caratteri via via che vengono digitati, è un display a cristalli liquidi (LCD). Contiene 8 righe di 40 caratteri ciascuna; ogni carattere è formato da una matrice di punti (6 x 8 punti). In TEXT, la colonna più a destra viene automaticamente lasciata vuota per salvaguardare la spaziatura tra le parole delle righe successive.

Il display può essere ruotato manualmente dalla posizione orizzontale ad una posizione inclinata (circa 30 gradi), per migliorarne la visibilità. Inoltre si può effettuare la regolazione del contrasto per mezzo di una rotellina posta sul lato destro di M10.

Si possono visualizzare fino a 8 righe di testo contemporaneamente. Operando in TEXT si possono eseguire "scroll" verso l'alto o verso il

basso utilizzando i tasti di spostamento cursore. In questa modalità è attiva la prestazione di scorrimento automatico delle parole: se una parola tende a superare la fine della riga, viene automaticamente spostata su quella successiva per evitare suddivisioni arbitrarie.

Sebbene il display di M10 presenti soltanto 40 caratteri per riga è possibile, quando si utilizza la stampante collegata, ottenere testi contenenti fino a 132 caratteri per riga.

CONNESSIONI DEL PANNELLO POSTERIORE

Sul pannello posteriore di M10 vi sono diversi connettori, come è illustrato nella Figura 1-8.

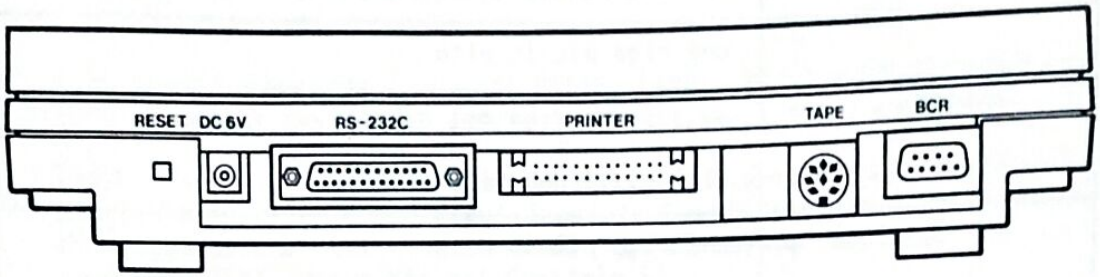

Fig. 1-8 Connettori del pannello posteriore

Essi sono, da sinistra verso destra :

RESET (azzeramento)

Pulsantino rientrato utilizzabile, come il nome stesso implica, per azzerare lo stato del computer, allorché lo stesso è disabilitato e tutte le funzioni sono bloccate. Ciò è piuttosto improbabile quando si utilizza uno dei programmi applicativi integrati, ma potrebbe accadere quando si provano per le prime volte programmi appena scritti.

DC 6V (corrente continua a 6 volt)

Spina per il collegamento dell'adattatore a corrente alternata, per alimentare M10 dalla rete pubblica.

RS-232C

Connettore di interfaccia conforme alle norme EIA RS-232C, ad eccezione del filo di rilevazione della portante (DCD) che non è connesso. Utilizzazioni tipiche di questa interfaccia sono:

- collegamento di M10 a modem esterno o accoppiatore acustico con modem integrato;

- collegamento di M10 ad apparecchiature che agiscono come terminali dati, quali le macchine per scrivere elettroniche o le stampanti seriali Olivetti;

- collegamento di M10 ad un altro M10 o ad altri Personal Computer.

PRINTER (stampante)

Connettore di interfaccia parallela di uscita, per il collegamento a M10 di stampanti o microplotter, come ad es. il modello Olivetti PL10.

TAPE (nastro)

Connettore per il collegamento eventuale di registratore di nastro a cassetta (consigliabile per memorizzare file). Ulteriori informazioni sono contenute nel Capitolo 9.

BCR

Presa per collegare a M10 un lettore di codici a barre (Bar Code Reader). E' stata progettata specificamente per i lettori di codici a barre Hewlett-Packard HEDS-3050 o HEDS-3000.

PANNELLO INFERIORE

Visto da sotto, M10 si presenta come nella Figura 1-9.

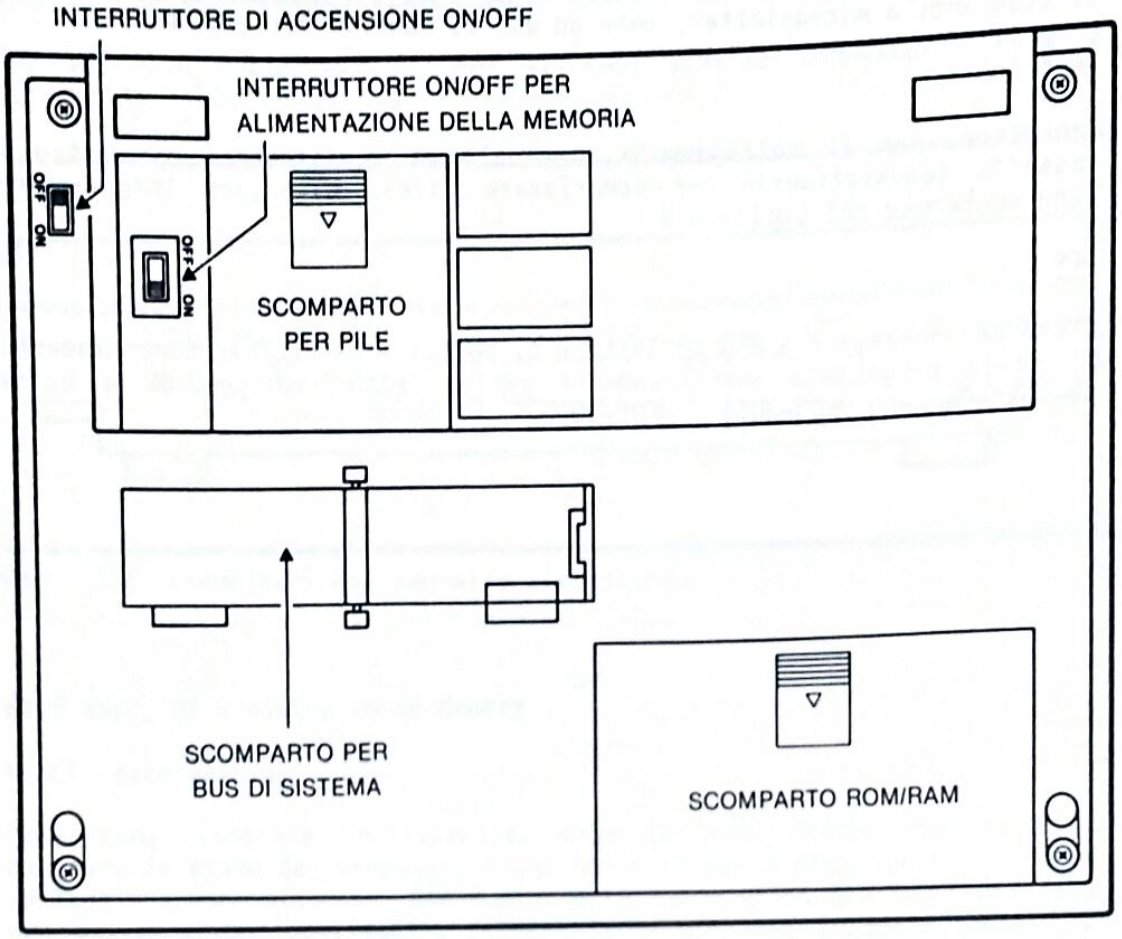

Fig. 1-9 Pannello inferiore

I componenti indicati sono:

1. Interruttore di accensione ON/OFF.

Accende (ON) o spegne (OFF) il computer.

2. Interruttore ON/OFF per alimentazione della memoria.

Consente di alimentare la memoria anche quando il computer è spento. Quando M10 viene consegnato, questo interruttore è in posizione OFF (spento). Durante la procedura di inizializzazione esso viene posizionato in ON e vi rimarrà finchè il computer non debba restare inutilizzato per un lungo periodo.

E' importante notare che, quando l'interruttore viene messo in OFF, i file contenuti nella memoria RAM di M10 vanno perduti. Tutti i file da conservare devono pertanto essere registrati su nastro prima di procedere allo spegnimento della memoria. Per prevenire spostamenti accidentali dell'interruttore, che potrebbero causare la perdita di informazioni valide e bloccare il funzionamento di M10, esso è inserito in una sede rientrata.

Le batterie ricaricabili garantiscono il mantenimento dei dati in memoria, a macchina spenta, per un periodo variabile da 8 a 30 giorni, in funzione della capacità di memoria installata. E' buona norma utilizzare o accendere M10 con regolarità, per mantenere cariche le batterie.

3. Scomparto per pile.

M10 è alimentato da quattro pile alcaline a secco da 1,5 V (AA), che devono essere inserite in questo scomparto secondo le indicazioni riportate nella Figura 2-1. Le pile hanno una durata di circa 20 ore, indipendentemente dalla capacità di memoria installata. Quando la tensione di alimentazione diventa troppo bassa, si accende un segnalatore luminoso sul pannello frontale (identificato dalla scritta "Battery Low"). Rimangono a questo punto circa 20 minuti di autonomia, prima che le pile diventino inutilizzabili. Spegnere il sistema e procedere alla sostituzione delle pile.

4. Scomparto ROM/RAM.

E' previsto per alloggiare le eventuali estensioni di memoria ROM e RAM. M10 ha di norma 32K byte di ROM, estendibili a 64K mediante aggiunta di un singolo componente. La capacità di RAM di base è di 8K byte, e può essere portata ad un massimo di 32K con moduli aggiuntivi di 8K ciascuno.

5. Scomparto bus di sistema.

Sotto il coperchio a cerniera si trova un connettore a 40 piedini, collegato al bus di sistema di M10.

2. ATTIVAZIONE DI M10

Oltre che con pile, M10 può essere alimentato da una batteria a 6 V (corrente continua), oppure collegato alla rete pubblica attraverso un adattatore per corrente alternata. Quest'ultimo accessorio, opzionale, può essere acquistato presso i concessionari e rivenditori Olivetti.

INSTALLAZIONE DELLE PILE

Essendo portatile, M10 è stato progettato per essere alimentato da pile (quattro pile alcaline a secco da 1,5 V - dimensione AA). Lo scomparto per le pile è nella parte inferiore della macchina, come mostrato nella Figura 1-9. La Figura 2-1 illustra come orientare e inserire le pile. Esse hanno una durata media di circa 20 ore di funzionamento, indipendentemente dalla quantità di memoria installata; dopo questo tempo l'indicatore luminoso 'Battery Low' posto sul pannello frontale si accenderà. Quando ciò accade, rimangono ancora circa 20 minuti di autonomia prima che le pile si esauriscano, quindi si dovrà procedere al più presto alla loro sostituzione.

Oltre alle pile, un accumulatore al Ni-Cd, posto all'interno di M10, alimenta la memoria anche quando la macchina è spenta. Questo accumulatore si ricarica automaticamente ogni volta che M10 è attivato e, quando sia completamente carico, può alimentare la memoria per un periodo variabile da otto a trenta giorni, secondo la quantità di memoria installata. La vita dell'accumulatore è di circa due anni; dopo tale periodo sarà necessaria la sostituzione, che potrà essere effettuata presso un concessionario o rivenditore autorizzato Olivetti.

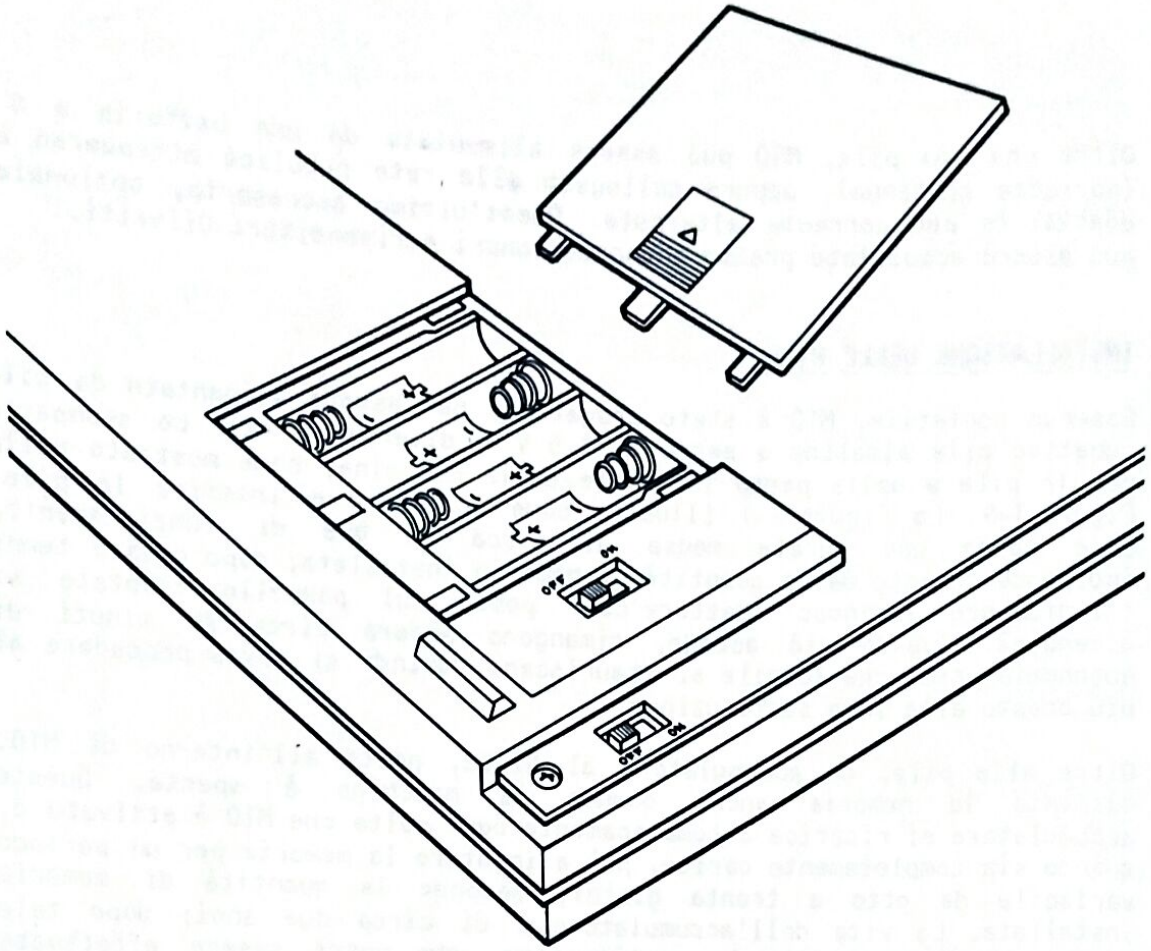

Fig. 2-1 Installazione delle pile

PRIMA ATTIVAZIONE

1. Togliere il coperchio dello scomparto delle pile, posto nella parte inferiore di M10.

2. Inserire quattro pile alcaline a secco da 1,5 V , di dimensioni AA, come illustrato in Figura 2-1.
   In alternativa, collegare la terminazione del cavo dell'adattatore alla spina DC 6V sul pannello posteriore di M10; indi collegare l'adattatore alla rete di alimentazione.
   E'importante ricordare che sia M10 che le periferiche devono essere spenti prima di collegare o scollegare l'adattatore a M10.

3. Porre l'interruttore di alimentazione della memoria ON/OFF in posizione ON (sulla parte inferiore, vedere Figura 1-9).

4. Porre l'interruttore ON/OFF di alimentazione in posizione ON (sulla parte inferiore, vedere Figura 1-9).

E' importante non dimenticare il passo 3. , altrimenti il computer non potrà funzionare.

Per spegnere M10, spegnere prima tutte le periferiche eventualmente collegate (ad es. la stampante) e porre in OFF l'interruttore principale. Non porre invece in OFF l'interruttore di alimentazione della memoria, perchè in questo caso tutti i suoi contenuti sarebbero cancellati.

Una volta eseguita la procedura di accensione iniziale, M10 deve essere acceso o spento agendo solamente sull'interruttore di alimentazione posto nella parte inferiore.

Va notato che M10 si spegnerà da solo, se non si preme alcun tasto per circa 10 minuti. Ciò permette di ottenere una maggiore durata delle pile. Per riprendere le operazioni, occorrerà in questo caso spegnere e riaccendere di nuovo M10. La funzione di spegnimento automatico può essere disabilitata tramite il comando BASIC "POWER CONT" o, in alternativa, si può modificare il tempo dopo cui M10 si spegne per mezzo del comando BASIC "POWER". Maggiori chiarimenti su questi comandi si trovano nella parte II del manuale

3. IL MENU PRINCIPALE

Quando si accende M10 a inizio lavoro, il display apparirà come nella Figura 3-1. Questo è il menù principale del computer.

```
13 Jun ,1983 Mon 11:46:07    (C)Microsoft
BASIC        TEXT        TELCOM      ADDRSS
SCHEDL       -.-         -.-         -.-
-.-          -.-         -.-         -.-
-.-          -.-         -.-         -.-
-.-          -.-         -.-         -.-
-.-          -.-         -.-         -.-
Select: _                29638 Bytes free
```

Fig. 3-1 Il menù principale

Si può vedere che il display contiene 8 righe di testo. Il rettangolo scuro posizionato su BASIC è il cursore, che può essere spostato facendo uso dei tasti di controllo descritti nel Capitolo 1. La parola su cui il cursore è posizionato appare "in negativo". Il contrasto può essere modificata facendo uso dell'apposito controllo situato a destra.

La prima riga del display è occupata dalla data e dall'ora. La prima volta su questa riga si leggerà:

01 Jan ,1900 Sun 00:00:00 (C) Microsoft

Le successive 6 righe sono riservate a un elenco dei programmi presenti nel computer. All'inizio vi saranno soltanto i cinque programmi applicativi integrati:

- BASIC
- TEXT
- TELCOM
- ADDRSS
- SCHEDL

C'è ancora sufficiente spazio sul menù per visualizzare il nome di altri 19 file.

Nell'ultima riga appare la scritta:

Select : _ 29638 Bytes free

L'opzione **Select** permette all'utente di intodurre il nome e accedere al programma o al file che desidera. Ciò può essere ottenuto anche posizionando il cursore sul nome del file scelto sul menù e premendo quindi il tasto <ENTER>.

L'informazione **29638 Bytes free** indica la quantità di memoria a disposizione dell'utente. All'attivazione questo valore dipende dalla memoria installata. I diversi valori sono elencati qui sotto:

Memoria installata	Byte liberi
32K	29638
24K	21446
16K	13254
8K	5062

Questi valori naturalmente diminuiranno quando si comincerà ad utilizzare la memoria del computer per registrarvi dati o programmi.

All'inizio il cursore è posizionato sul programma BASIC, ma può essere spostato agendo sui tasti movimento cursore, indicati dalle frecce. All'attivazione sono tuttavia disponibili solamente BASIC, TEXT e TELCOM. Le procedure per attivare ADDRSS e SCHEDL sono descritte nei Capitoli 6 e 7 rispettivamente.

IMPOSTAZIONE DELL'ORA

Come già detto, all'attivazione compaiono sulla prima riga del menù principale la data, il giorno e l'ora nella forma:

01 Jan , 1900 Sun 00:00:00

e da questa data M10 inizia la misura del tempo.

La procedura per impostare i valori reali è descritta nel seguito.

REIMPOSTAZIONE DELL'ORA:

Posizionare il cursore su BASIC e premere <ENTER>. Il cursore apparirà come un rettangolo scuro lampeggiante sotto il messaggio tipico del BASIC, **Ok**.

Impostare sul display:

TIME$="ora:minuti:secondi"

e premere <ENTER>, dove:

ora, è un numero di due cifre da 00 a 23

minuti, è un numero di due cifre da 00 a 59

secondi, è un numero di due cifre da 00 a 59

in formato identico a quello di un orologio digitale.

L'orologio interno di M10 registrerà quest'ora nel momento in cui è premuto il tasto <ENTER>.

Se l'ora impostata non è in formato corretto verrà visualizzato un messaggio d'errore.

Per verificare il valore impostato, digitare:

PRINT TIME$

e premere <ENTER>.

L'ora verrà visualizzata sul display.

REIMPOSTAZIONE DELLA DATA:

Digitare la data nel seguente modo:

DATE$="giorno/mese/anno"

e premere <ENTER>, dove

giorno, è un numero di due cifre da 01 a 31

mese, è un numero di due cifre da 01 a 12

anno, è un numero di due cifre da 00 a 99

Ad esempio, l'istruzione DATE$="14/06/83" seguita da <ENTER>, serve a impostare la data 14 giugno 1983.

Il comando PRINT DATE$ visualizza sul display la data impostata.

REIMPOSTAZIONE DEL GIORNO:

Impostare il giorno della settimana come segue:

DAY$="giorno"

e premere <ENTER>, dove 'giorno' è una delle seguenti abbreviazioni di tre lettere:

Lunedì	- Mon
Martedì	- Tue
Mercoledì	- Wed
Giovedì	- Thu
Venerdì	- Fri
Sabato	- Sat
Domenica	- Sun

Anche in questo caso il valore impostato può essere verificato con il comando PRINT DAY$.

Per ritornare al menù principale, premere il tasto funzione <F8>. Il giorno, la data e l'ora appariranno sulla prima riga del menù.

4. L'AMBIENTE BASIC

Il BASIC è uno dei cinque programmi integrati di cui M10 è corredato. La maggior parte delle informazioni richieste per sfruttare al meglio questa prestazione sono contenute nella Parte II del manuale (la guida di riferimento del linguaggio BASIC). Per tutti i problemi relativi alla programmazione, alla sintassi e alla struttura del linguaggio, si potrà ricorrere a questa guida, che contiene anche una lista completa di tutti i comandi BASIC, con accenni al loro funzione ed uso. Tuttavia, dal momento che il BASIC è il punto centrale di M10, questo capitolo ne sottolinea gli aspetti fondamentali, per maggior completezza.

BASIC sta per "Beginners All-purpose Symbolic Instruction Code" (codice di istruzioni simboliche di uso generale per principianti), ed è il linguaggio di programmazione ad alto livello di M10. Con questa prestazione l'utente può scrivere, memorizzare ed eseguire i suoi programmi applicativi, scritti in BASIC.

IL DISPLAY E LE MODALITA' OPERATIVE

Per selezionare il BASIC dal menù principale, posizionare il cursore sul nome BASIC e premere <ENTER>. Il display si configura come illustrato nella Figura 4-1. Il cursore appare come un rettangolo lampeggiante, immediatamente sotto il messaggio BASIC, **Ok** .

In alternativa, si può caricare un programma BASIC già esistente ed eseguirlo, posizionando il cursore sul nome del file BASIC (tutti i file BASIC hanno il suffisso .BA) e premendo <ENTER>.

```
OLIVETTI M10 BASIC 1.0
(C) 1983 Microsoft
29382 Bytes free
Ok
▒

File Load Save Run  List                Menu
```

Fig. 4-1 Display ottenuto selezionando BASIC

Il BASIC ha tre modalità operative:

- Stato Comandi
- Stato Esecuzione
- Stato Editor di Linea

Non è necessario che l'utente selezioni le prime due modalità: il computer lo fa automaticamente in funzione delle operazioni da eseguire.

STATO COMANDI

Questa è la modalità attiva quando sul display compare il messaggio **Ok** . In Stato Comandi è possibile impostare linee di programma o linee ad esecuzione immediata (il concetto di linea è spiegato più sotto). Quando si preme il tasto <ENTER> per eseguire il programma, M10 passa in Stato Esecuzione, per ritornare in Stato Comandi al termine dell'operazione.

STATO ESECUZIONE

Questa è la modalità in cui M10 si trova durante l'esecuzione di un programma o di una linea immediata. Al termine esso ritorna in Stato Comandi e visualizza nuovamente il messaggio **Ok**.

STATO EDITOR DI LINEA

Questa modalità viene richiamata col comando EDIT <ENTER>, in presenza del messaggio BASIC **Ok** sul display. Viene utilizzata per correggere o modificare con facilità le linee di un programma. La funzione dei tasti di movimento cursore in questo caso è descritta nel Capitolo 5. Per uscire dallo Stato Editor di Linea, e tornare in Stato Comandi, occorre premere il tasto <F8>.

Il concetto di linea è il seguente:

. Una linea 'logica' è una linea di comandi o istruzioni, che termina con un ritorno a capo (tasto <ENTER>), e con lunghezza massima di 255 caratteri.

. Una linea 'fisica' è una linea di testo sul display, e perciò non può superare i 40 caratteri di lunghezza.

. Una linea 'immediata' è un comando, o serie di comandi, eseguiti immediatamente dal computer una volta premuto <ENTER>, ad es.:

PRINT TIME$

Una linea immediata non può iniziare con un numero: il computer la interpreterebbe come un programma o parte di un programma.

TASTI FUNZIONE IN AMBIENTE BASIC

Le tabelle che seguono riassumono il significato assunto dai tasti funzione F1-F8, PASTE, LABEL e PAUSE/BREAK, quando si usa il programma BASIC.

NOME TASTO	FUNZIONE
F1 File	elenca sul display tutti i file del menù
F2 Load	carica un file da memoria o da unità esterna (es.:registratore di nastro a cassette)
F3 Save	registra il programma attuale in memoria o su unità esterna, come ad es. un registratore di nastro a cassette
F4 Run	esegue il programma attuale
F5 List	lista il programma corrente su display
F6	non utilizzato
F7	non utilizzato
F8 Menu	ritorna al menù principale
PASTE	inserisce il contenuto del buffer PASTE alla posizione in cui si trova il cursore
LABEL	visualizza le funzioni di F1-F8 sul display
PAUSE	arresta momentaneamente il programma; l'esecuzione riprende premendo nuovamente lo stesso tasto
BREAK (SHIFT+PAUSE)	arresta l'esecuzione del programma

Fig. 4-2 Tasti funzione in Stato Comandi

E' anche possibile in un programma definire secondo le proprie esigenze il significato dei tasti funzione. La procedura per ottenerlo viene spiegata nella seconda parte di questo manuale.

NOME TASTO	FUNZIONE
F1 File	permette di definire una stringa da ricercare nel file
F2 Load	carica un programma da un registratore di nastro a cassette
F3 Save	registra un programma su un registratore di nastro a cassette
F4	non utilizzato
F5 Copy	memorizza una stringa selezionata nel buffer PASTE, per permetterne la copiatura nel punto indicato dal cursore
F6 Cut	memorizza una stringa selezionata nel buffer PASTE, e rimuove l'originale dal file
F7 Sel	permette di selezionare una stringa, per eseguire operazioni di COPY, CUT e PASTE
F8 Exit	per uscire dallo Stato Editor di Linea
PASTE	inserisce il contenuto del buffer PASTE alla posizione in cui si trova il cursore
LABEL	visualizza le funzioni di F1-F8 sul display

Fig. 4-3 Tasti funzione in Stato TEXT o Editor di Linea

CARATTERI DI CONTROLLO IN BASIC

In BASIC sono operanti i seguenti caratteri di controllo:

<CTRL> + C Arresta l'esecuzione di un programma lasciando M10 in BASIC.

<CTRL> + H Cancella il carattere precedente.

<CTRL> + I Comando di tabulazione.

<CTRL> + M Comando di ritorno a capo.

<CTRL> + S Arresta temporaneamente l'esecuzione di un programma; essa può essere ripresa premendo nuovamente <CTRL> + S.

<CTRL> + U Cancella la linea attuale.

<CTRL> + X Cancella la linea attuale.

5. IL PROGRAMMA APPLICATIVO TEXT

Il secondo dei cinque programmi integrati elencati nel menù è denominato TEXT. Si tratta di un programma per la gestione dei testi, progettato appositamente per la creazione, l'impaginazione e la memorizzazione dei file testo. Esso offre una buona gamma di funzioni "word processing": funzioni di ricerca, inserimento e cancellazione di brani di testo, operazioni di "taglia e incolla", etc. La prima opportunità di impiego del programma TEXT da parte dell'utente, si verifica probabilmente nella generazione dei file ADRS.DO e NOTE.DO , richiesti nelle applicazioni ADDRSS e SCHEDL (come descritto nei Capitoli 6 e 7).

Per accedere a TEXT si posiziona il cursore sul nome TEXT all'interno del menù principale e si preme <ENTER>. In alternativa è possibile utilizzare il comando:

TEXT<ENTER>

inviato direttamente da tastiera. Esso apparirà all'estremità inferiore dello schermo, alla destra di **Select:** . Premendo <ENTER>, lo schermo appare come in Figura 5-1.

```
File to edit? ▒
```

Fig. 5-1 Il display per il programma TEXT

NOMI DEI FILE NEL PROGRAMMA TEXT

In risposta al messaggio:

File to edit?

l'utente deve introdurre il nome del file da elaborare o creare e premere <ENTER>. Vi sono alcuni aspetti da tenere a mente nella scelta del nome per un file di M10. Il nome può avere lunghezza massima di 6 caratteri, con le seguenti restrizioni:

- Un nome di file non può iniziare con un numero.

- I seguenti simboli non sono accettati come primo carattere di un nome di file:

  ! " £ $ % & ' *

  ( ) + : , . /

- Inoltre il punto (.) e i due punti (:) sono proibiti in un nome di file. L'unica eccezione si ha quando il punto si trova all'estrema destra del nome del file, nel qual caso viene incorporato nel suffisso (vedere esempio sottostante).

- M10 converte tutte le lettere in maiuscole, cosicchè non è possibile usare le maiuscole per distinguere nomi di file altrimenti identici.

Dopo aver scritto il nome del file selezionato, premere <ENTER>. Se il nome viene rifiutato per violazione delle regole citate, M10 emette un segnale acustico di avvertimento e ripropone il messaggio **File to edit?** sulla riga successiva. Se è accettato, il display è cancellato. Nell'angolo in alto a sinistra appare un cursore contente una freccia rivolta a sinistra, per indicare l'inizio del testo.

I nomi dei file creati in fase di TEXT sono completati con il suffisso .DO da parte del computer. Non è necessario che l'utente lo specifichi. Se, ad esempio, si specifica il nome PIPPO in risposta alla richiesta, il file verrà listato come PIPPO.DO nel menù principale. Gli esempi che seguono presentano alcuni casi di nomi leciti e illeciti.

Nome impostato	Nome allocato da TEXT
PIPPO	PIPPO.DO
pippo	PIPPO.DO
pIPPo	PIPPO.DO
pippo.	PIPPO.DO
P I P P O	P I P P O.DO
$PIPPO	Non accettato
PIPPO$	PIPPO$.DO
PIP.PO	Non accettato
"PIPPO"	Non accettato
PIPPO.DO	PIPPO.DO

Nella RAM possono essere memorizzati fino a un massimo di 19 file alla volta; il loro nome apparirà sul menù in aggiunta a quello dei cinque programmi integrati. Se se ne dovessero aggiungere altri, alcuni di quelli presenti dovranno essere memorizzati su dispositivo esterno, come ad esempio un registratore di nastro a cassette (vedere Capitolo 9).

Per cancellare un file dalla RAM, si deve richiamare il programma BASIC dal menù principale e impostare quindi il seguente comando:

KILL "nomefile" <ENTER>

E' importante ricordare le doppie virgolette all'inizio e alla fine del nome del file. E' altrettanto importante digitare interamente il nome, compreso il suffisso.

SCRITTURA DI UN FILE TESTO

Una volta attribuito un nome valido al file, si può iniziare a digitare il testo sulla tastiera. Esso viene visualizzato così come è digitato. Come già segnalato, lo schermo contiene 40 caratteri per riga (linea fisica). Non è necessario premere <ENTER> per andare a capo (a meno che non sia richiesto, come ad es. alla fine di un paragrafo); il cursore procede automaticamente di riga in riga, riposizionando le parole quando necessario. Per correggere errori di battitura, posizionare il cursore uno spazio a destra della lettera da cancellare e premere <DEL/BS>. Premendo <SHIFT> + <DEL/BS>, si cancella il carattere su cui il cursore è posizionato. Per inserire un carattere o un blocco di testo, è sufficiente portare il cursore nella posizione desiderata e digitare l'inserto; il testo che segue viene spostato verso destra di uno spazio per ciascuno dei caratteri inseriti. M10 opera permanentemente in modalità 'inserimento'. Per eseguire un'aggiunta al testo contenuto in un file, si posiziona il cursore al termine del testo stesso e si introduce la parte aggiuntiva.

La posizione del cursore è controllata dai tasti di movimento cursore, cioé il gruppo di quattro tasti identificati da frecce e situati nella parte superiore destra della tastiera, appena sotto lo schermo. La loro funzione, già descritta nel Capitolo 1, viene qui ripresa per comodità. ed illustrata dettagliatamente in Figura 5-2.

Come la maggior parte dei tasti, anche quelli per il movimento cursore, se mantenuti premuti, eseguono la funzione con ripetizione. Tenendo premuto il tasto con la freccia a destra, il cursore esegue una scansione da sinistra verso destra, dall'estremità di una riga all'inizio della successiva. Quando raggiunge l'estremità inferiore dello schermo, la scansione continua facendo apparire la linea successiva. Il tasto con la freccia a sinistra esegue la scansione in senso inverso, da destra verso sinistra e dal basso verso l'alto, in maniera del tutto analoga. Tenendo premuto il tasto con la freccia verso l'alto, il cursore si sposta verticalmente verso la sommità dello schermo, facendo apparire la parte di testo che precede, sino a raggiungere l'inizio del file. Il tasto con la freccia verso il basso svolge la funzione analoga nella direzione opposta, sino a raggiungere la fine del file. Questi tasti hanno, come si dice, una funzione di "scrolling" (scorrimento all'interno del testo).

TASTO	MOVIMENTO CURSORE
→	uno spazio a destra
<SHIFT> + →	all'inizio della parola seguente
<CTRL> + →	alla fine della riga in corso
←	uno spazio a sinistra
<SHIFT> + ←	all'inizio della parola precedente (o attuale)
<CTRL> + ←	all'inizio della riga in corso
↑	una riga più in alto
<SHIFT> + ↑	alla prima riga del display
<CTRL> + ↑	all'inizio del file
↓	una riga più in basso
<SHIFT> + ↓	all'ultima riga del display
<CTRL> + ↓	alla fine del file

Fig. 5-2 Tasti di movimento cursore

TASTI FUNZIONE E COMANDO IN TEXT

In modalità TEXT i tasti F1-F8 hanno speciali funzioni di controllo (vedere Figura 5-3).

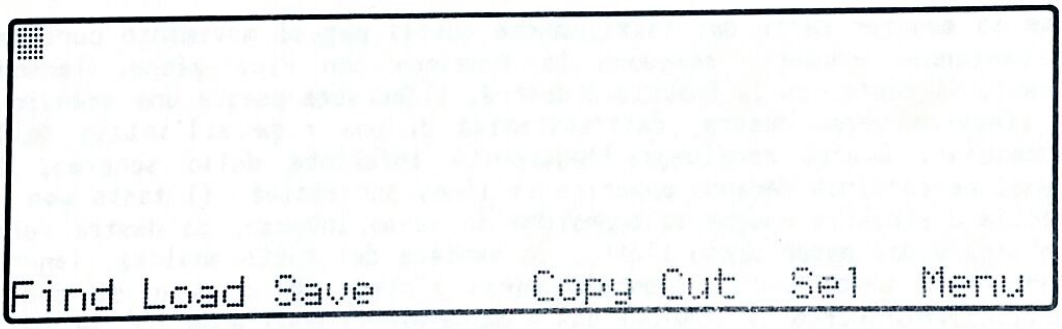

Fig. 5-3 Visualizzazione del significato dei tasti funzione

Per visualizzarle sul display, si deve premere <LABEL>. Esse offrono

all'utente una vasta gamma di prestazioni di "word-processing", che sono descritte nei paragrafi seguenti. Una sintesi delle funzioni e tasti comando è inoltre riportata in Figura 5-6.

RICERCA DI UNA STRINGA DI CARATTERI IN TEXT

La funzione 'Find' (cerca) è attivata dal tasto funzione <F1>. Per selezionarla basta premere questo tasto. Sul display comparirà il messaggio

String:

al fondo dello schermo, con il cursore immediatamente a destra. Digitare la stringa di caratteri che deve essere ricercata e premere <ENTER>. Normalmente si tratterà di una parola o di una frase. Il cursore si muoverà per indicare il punto del testo dove la stringa compare per la prima volta ed il messaggio scomparirà dal display. Premendo ancora <F1> il messaggio e la stringa di ricerca riappaiono. Con <ENTER> il programma ricerca la successiva presenza della stessa stringa, procedendo in avanti nel testo. Se la stringa non è più presente nel testo, viene visualizzato il messaggio:

No match

Vi sono alcuni punti importanti da ricordare quando si utilizza questa prestazione.

1. La stringa non può superare i 24 caratteri, compresi gli spazi. Se si tenta di superare questo limite il computer emette una segnalazione acustica di avvertimento e il cursore rimane fermo.

2. La spaziatura è di vitale importanza. Se la spaziatura impostata è errata la stringa non potrà essere trovata.

3. La ricerca inizia a partire dalla posizione attuale del cursore. Se si desidera effettuare la ricerca su tutto il testo, bisogna assicurarsi che il cursore si trovi all'inizio del file prima di premere <F1>. Non c'è modo di far eseguire la ricerca all'indietro.

4. In questa applicazione, come in altre, M10 non distingue tra lettere maiuscole e minuscole e ignora quindi queste differenze quando confronta la stringa di ricerca con il testo.

Per cambiare stringa di ricerca, premere <F1> e digitare la nuova stringa. Quest'ultima elimina la precedente.

LA FUNZIONE 'LOAD' IN TEXT

Questa funzione, richiamata premendo il tasto <F2>, permette di caricare informazioni da un registratore di nastro a cassetta in un file TEXT. La procedura è la seguente:

1. Assicurarsi che M10 sia collegato ad un registratore di nastro adatto (per i dettagli consultare il Capitolo 9).

2. Accedere al file TEXT dove dovranno essere caricati i dati (se necessario creare un nuovo file, come descritto all'inizio del capitolo).

3. Premere <F2>. A questo punto viene visualizzato il messaggio:

Load from:

4. In risposta, digitare il nome del file da caricare e premere <ENTER>. Mentre ricerca il file specificato sul registratore, il computer emette un suono acuto. Non appena il file è stato trovato, sul display viene visualizzato il messaggio:

FOUND: nomefile

dove 'nomefile' è quello digitato in precedenza. Se sulla cassetta sono registrati più file, ogni volta che nel corso della ricerca ne viene incontrato uno non corrispondente a quello ricercato comparirà il messaggio:

SKIP: nomefile

Se si specifica che l'informazione presente sulla cassetta deve essere caricata su un file TEXT che già contiene un testo, il contenuto del file ricercato sulla cassetta verrà messo in coda al testo già esistente.

Maggiori dettagli sull'utilizzo del registratore di nastro a cassetta collegato a M10 sono riportati nel Capitolo 9.

LA FUNZIONE 'SAVE' IN TEXT

Dopo aver creato o modificato un file TEXT, può essere necessario archiviarlo in un file su cassetta, utilizzando la funzione 'Save' (F3). Per far questo occorre procedere come segue:

1. Assicurarsi che M10 sia collegato a un registratore di nastro adatto (per i dettagli consultare il Capitolo 9).

2. Accedere al file TEXT che si vuole registrare ('salvare') su cassetta e premere <F3>. Il messaggio:

Save to:

apparirà sull'ultima riga del display.

3. Scegliere un nome di file (non più lungo di sei caratteri) per il file su cassetta nel quale deve essere archiviato il file TEXT.

4. Digitare questo nome e premere <ENTER>.

Quando il messaggio scompare dallo schermo, il file è stato registrato sotto il nome scelto.

Maggiori dettagli sull'utilizzo del registratore di nastro a cassetta

collegato a M10 sono riportati nel Capitolo 9.

LA FUNZIONE 'SELECT'

La funzione 'Select' (scegli) permette di individuare un blocco di testo, su cui effettuare successivamente operazioni di copiatura ('Copy' - F5), taglio ('Cut' - F6) o spostamento (Cut and Paste, ovvero il taglia e cuci elettronico). Tale blocco può variare in lunghezza da un singolo carattere all'intero testo del file.

Vi sono comunque dei limiti, dati dalla quantità di memoria disponibile. Se il blocco selezionato ha dimensione maggiore di quella indicata nel menù principale come libera (**Bytes free**), il sistema visualizzerà il messaggio di memoria piena:

Memory full

Analogamente, se si riceve questo messaggio una volta impostato il comando PASTE, significa che si è ecceduta la capacità di memoria RAM. Occorre quindi prestare particolare attenzione quando si opera su brani di testo lunghi.

Per definire un blocco, posizionare il cursore all'inizio del testo da individuare e premere <F7>. Quindi portare il cursore alla fine del blocco, utilizzando i tasti di movimento cursore, sia carattere per carattere - con la freccia a destra - che riga per riga - con la freccia verso il basso -. I caratteri che in tal modo vengono inclusi nel blocco selezionato appaiono 'in negativo' (scritte bianche su fondo nero), come illustrato in Figura 5-4.

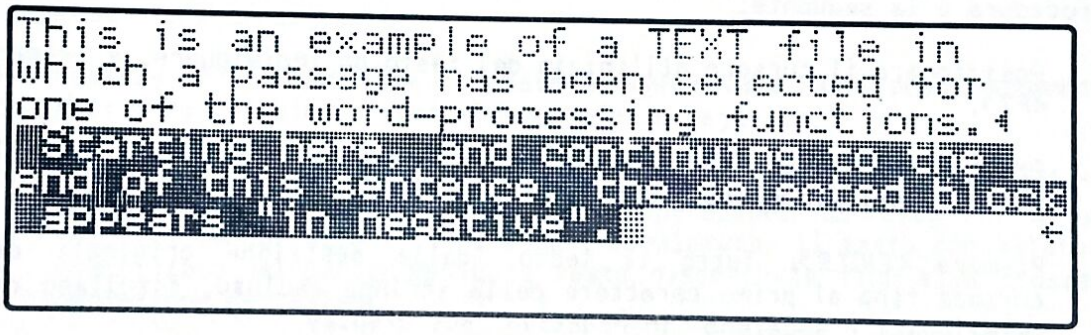

Fig. 5-4 Un blocco di testo selezionato

In pratica il testo può essere definito, a partire dalla posizione del cursore, in avanti o all'indietro lungo il file. Ciò è sintetizzato in Figura 5-5.

TESTO DA SELEZIONARE	SEQUENZA DI TASTI
Carattere seguente	<F7> poi <Freccia destra>
Carattere precedente	<F7> poi <Freccia sinistra>
Parola seguente	<F7> poi <SHIFT> + <Freccia destra>
Parola precedente	<F7> poi <SHIFT> + <Freccia sinistra>
Da cursore a fine riga	<F7> poi <CTRL> + <Freccia destra>
Da cursore a inizio riga	<F7> poi <CTRL> + <Freccia sinistra>
Da cursore a fine display	<F7> poi <SHIFT> + <Freccia in basso>
Da cursore a inizio display	<F7> poi <SHIFT> + <Freccia in alto>
Da cursore a fine file	<F7> poi <CTRL> + <Freccia in basso>
Da cursore a inizio file	<F7> poi <CTRL> + <Freccia in alto>

Fig. 5-5 Come definire un blocco di testo

Si può selezionare un blocco di testo anche utilizzando la funzione 'Find' (trova). Essa permette di definire un testo a partire dalla posizione attuale del cursore fino ad una particolare parola o frase. La procedura è la seguente:

1. Posizionare il cursore all'inizio del testo da individuare, e premere <F7>.

2. Premere <F1> e digitare la stringa di caratteri o la parola alla quale la selezione del blocco deve terminare.

3. Premere <ENTER>. Tutto il testo, dalla posizione originale del cursore fino al primo carattere della stringa escluso, risultano ora selezionati e appaiono 'in negativo' sul display.

Se fosse stato selezionato un blocco di testo maggiore di quello desiderato, si potrà rimuovere la parte eccedente spostando il cursore all'indietro fino al punto finale desiderato. Per annullare un'operazione di 'Select' in corso, premere <BREAK> (cioè <SHIFT> + <PAUSE/BREAK>): il blocco in fase di selezione riassumerà l'assetto precedente.

UTILIZZO DELLA FUNZIONE 'COPY'

Il tasto funzione F5 permette di copiare (duplicare) un blocco di testo, ad esempio una parola o una frase di uso frequente, in un'altra parte del file. Le funzioni 'Copy' e 'Cut' comportano l'utilizzo non solo dei rispettivi tasti funzione (F5 ed F6), ma anche della funzione 'Select' e del comando PASTE. Per eseguire un'operazione di 'Copy' procedere come segue:

1. Premere <F7> e definire il blocco di testo da copiare, come descritto in precedenza. Il blocco apparirà 'in negativo' sul display.

2. Premere <F5>. Il fondo scuro scompare, indicando così che il blocco selezionato è stato copiato nel buffer PASTE.

3. Portare il cursore nella posizione in cui la copia deve essere inserita e premere <PASTE>.

Si noti che l'inserimento avviene in modo che l'ultimo carattere della copia vada a occupare la posizione che precede il cursore. Solo se l'ultimo carattere del blocco selezionato è un 'ritorno a capo' (ottenuto con <ENTER>) si posizionerà sulla riga precedente. Se il medesimo blocco deve essere copiato più volte nel testo, sarà sufficiente ripetere solo il passo 3, in quanto il blocco selezionato rimane nel buffer PASTE fino a che non viene sostituito in una operazione di selezione successiva.

LA FUNZIONE 'CUT' E 'PASTE'

Permette di cancellare un blocco, o di spostarlo da una posizione a un'altra del file. Per la cancellazione si utilizza la sola funzione 'taglia' (F6 CUT) dopo la scelta del blocco (F7 SELECT); per l'operazione di 'scegli, taglia e incolla' occorre utilizzare anche il tasto <PASTE>. La procedura è la seguente:

1. Premere <F7> e definire il testo da cancellare come precedentemente descritto. Il blocco prescelto appare in negativo.

2. Premere <F6>. Il blocco selezionato viene cancellato dal file e memorizzato nel buffer PASTE. Se l'operazione da eseguire è una semplice cancellazione, essa è così completata. Il testo cancellato è memorizzato nel buffer PASTE, e resta disponibile fino alla prossima operazione di 'Select'.

3. Per spostare invece il testo, occorre portare il cursore nella nuova posizione e premere <PASTE>. Il testo cancellato sarà inserito nel punto desiderato.

NOTA: Sia la procedura 'copia' che quella 'taglia e incolla' possono essere utilizzate per trasferire dati da un file ad un altro. La procedura da adottare è analoga a quella descritta. Invece di posizionare il cursore nel file attuale per l'inserimento, occorre accedere prima al file di destinazione, posizionare il cursore nel punto opportuno e quindi premere <PASTE>.

STAMPA DI UN FILE TEXT

Per stampare un file TEXT (testo), occorre utilizzare il tasto comando PRINT. Ovviamente, è necessario che M10 sia collegato a una stampante (tramite il cavo Olivetti per stampanti, da inserire nel connettore PRINTER sul pannello posteriore).

- Premendo <PRINT> viene stampato il contenuto del display.

- Premendo <SHIFT> + <PRINT> viene invece stampato l'intero file TEXT.

In quest'ultimo caso viene emesso il messaggio:

Width?

accompagnato dalla larghezza della riga di stampa attualmente definita. Se si desidera cambiarne il valore, digitare il nuovo numero (che dovrà essere compreso tra 10 e 132) e premere <ENTER>. Questa operazione definisce il numero di caratteri per riga del testo stampato.

Premendo <ENTER> senza impostare alcun valore, sarà assunto come valido l'ultimo valore impostato. Il valore assunto da M10 in mancanza di impostazioni precedenti (default), è di 80 caratteri per riga.

LA FUNZIONE 'MENU'

Il tasto <F8>, chiamato 'Menù', ha due funzioni contemporanee. La pressione di questo tasto infatti chiude il file e ritorna al menù principale. Inoltre il file viene memorizzato in RAM con il suo nome di file, che sarà pertanto presente sul display nel menù stesso.

SOMMARIO DEI TASTI FUNZIONE E COMANDO

La Figura 5-6 illustra sinteticamente l'utilizzo dei vari tasti funzione e comando quando si opera in modalità TEXT.

TASTO	FUNZIONE
F1 Find	Ricerca una stringa di caratteri nel testo.
F2 Load	Carica un file da cassetta su M10.
F3 Save	Registra un file su cassetta.
F4	Non utilizzato in TEXT.
F5 Copy	Usato, con il tasto PASTE, per copiare una parte di testo in un'altra posizione o in altro file.
F6 Cut	Taglia una parte di testo (per cancellarla); la memorizza nel buffer PASTE per eventuale utilizzo successivo (taglia e incolla)
F7 Select	Definisce un blocco di testo per successive operazioni.
F8 Menu	Chiude il file e riporta al menù principale.
PASTE	Inserisce il contenuto del buffer PASTE nella posizione indicata dal cursore.
LABEL	Visualizza le funzioni sul display.
PRINT	Stampa il contenuto del display o dell'intero file (SHIFT + PRINT) su stampante esterna.
PAUSE/BREAK	In modalità TEXT è utilizzato solo BREAK, per annullare le operazioni - Select - Find - Load - Save.

Fig. 5-6 Tasti funzione e comando in TEXT

CARATTERI DI CONTROLLO CORRISPONDENTI

Le azioni provocate dai tasti funzione e di movimento cursore possono essere anche ottenute con caratteri di controllo, generati dalla digitazione contemporanea di <CTRL> (controllo) e di un tasto alfabetico. Naturalmente non è indispensabile conoscere queste sequenze, in quanto si hanno a disposizione tasti specializzati, ma potrebbe essere utile in qualche caso conoscere la corrispondenza fra i due tipi di comandi. Tale corrispondenza è riassunta in Figura 5-7.

CARATTERI DI CONTROLLO	EQUIVALENZA
<CTRL> + A	<SHIFT> + <Freccia sinistra>
<CTRL> + B	<SHIFT> + <Freccia in basso>
<CTRL> + C	<BREAK>
<CTRL> + D	<Freccia destra>
<CTRL> + E	<Freccia in alto>
<CTRL> + F	<SHIFT> + <Freccia destra>
<CTRL> + G	<F3>
<CTRL> + H	<DEL/BS>
<CTRL> + I	< →I >
<CTRL> + L	<F7>
<CTRL> + M	<ENTER>
<CTRL> + N	<F1>
<CTRL> + Q	<CTRL> + <Freccia sinistra>
<CTRL> + R	<CTRL> + <Freccia destra>
<CTRL> + S	<Freccia sinistra>
<CTRL> + T	<SHIFT> + <Freccia in alto>
<CTRL> + U	<F6>
<CTRL> + V	<F2>
<CTRL> + W	<CTRL> + <Freccia in alto>
<CTRL> + X	<Freccia in basso>
<CTRL> + Y	<PRINT>
<CTRL> + Z	<CTRL> + <Freccia in basso>

Figura 5-7 Corrispondenza fra caratteri di controllo e tasti funzione

6. IL PROGRAMMA APPLICATIVO ADDRSS

Questo programma fornisce uno strumento per creare, aggiornare e ricercare facilmente, informazioni contenute in un indirizzario o agenda telefonica (ADDRSS = indirizzario elettronico). Le funzioni di ADDRSS, come viene spiegato in questo capitolo, permettono di soddisfare diverse esigenze di ricerca o di ottenere elenchi differenziati, in base al nome - al numero telefonico - alla categoria - alla professione - all'area geografica, ecc..

Per utilizzare il programma occorre per prima cosa procedere all'introduzione in memoria delle informazioni necessarie, in questo caso l'elenco dei nominativi, degli indirizzi e dei numeri telefonici che costituiranno l'indirizzario.

CREAZIONE DEL FILE 'ADRS.DO'

Accedere a TEXT, posizionando il cursore sul nome TEXT del menù principale e premendo <ENTER>. In risposta al messaggio, digitare il nome del file. Per questa particolare applicazione non c'è facoltà di scelta: il nome del file dovrà essere obbligatoriamente ADRS.DO.

Se si richiedesse il programma ADDRSS prima di aver creato il file ADRS.DO, verrebbe visualizzato il messaggio:

ADRS.DO not found	(ADRS.DO non trovato)
Press space bar for MENU	(premere barra spazio per MENU)

Una volta creato il file ADRS.DO, occorre introdurre i nomi, gli indirizzi e i numeri telefonici che si desiderano nel proprio elenco. Si può adottare qualunque formato che meglio risponda alle esigenze personali; tuttavia, una volta scelto un formato, è raccomandabile applicarlo con continuità. Ciò faciliterà molto la consultazione dell'indirizzario. Un formato come quello riportato nel seguito è perfettamente adeguato:

Nome :numero telefonico: indirizzo

Esso faciliterà le operazioni di ricerca (Find) e permetterà di ottenere elenchi di indirizzi immediatamente comprensibili, sia sul display che in stampa.

E' inoltre indispensabile che ogni record sia chiuso con <ENTER>. Questo è infatti l'unico modo con cui il computer può distinguere tra un record e il successivo. Il simbolo <ENTER>, una punta di freccia scura orientata a sinistra, non dovrà apparire in nessun altra parte che alla fine di ciascun record.

Si ricordi che ADRS.DO è un file di tipo TEXT, e che perciò tutte le

opzioni di 'editing' descritte nel Capitolo 5 sono utilizzabili per effettuarvi modifiche, inserimenti e sostituzioni. Inoltre deve essere chiara la differenza fra ADRS.DO e ADRRSS, due file correlati ma concettualmente molto diversi. ADRS.DO è il file contenente gli indirizzi; ADDRSS è invece il programma che ordina e organizza le informazioni secondo le istruzioni fornite dall'utente.

Quando la lista dei nomi, numeri telefonici e indirizzi è completa, si chiude il file e si ritorna al menù premendo <F8>. Il file apparirà sulla lista del menù con il nome ADRS.DO.

UTILIZZO DEL PROGRAMMA ADDRSS

Il programma ADDRSS può ora essere richiamato posizionando il cursore sulla voce ADDRSS e premendo <ENTER>.

Il display assumerà l'aspetto illustrato in Figura 6-1.

```
Adrs:

Find          Lfnd          Menu
```

Fig. 6-1 Display ottenuto selezionando ADDRSS

Sulla prima riga appare il messaggio:

Adrs: (address = indirizzo)

Come si può osservare, in questo programma sono utilizzati solo tre tasti funzione : F1, F5 e F8.

- F1 richiama la funzione 'Find' (trova).
- F5 (denominato 'Lfnd') ha la stessa funzione di F1, con la differenza che il risultato viene inviato in stampa.
- F8, come nelle altre applicazioni, fa ritornare al menù principale.

La funzione 'Find' ha un ruolo identico a quello svolto nel programma TEXT. Sul display compare la richiesta:

Adrs: (Indirizzo:)

Per ricercare un particolare record nel file ADRS, basta premere <F1> (ordine di ricercare) e digitare una stringa di caratteri dopo la parola 'Find' che viene conseguentemente visualizzata. Tale stringa può essere di lunghezza qualsiasi, da un solo carattere all'intero record.

Tutti i record contenenti la stringa digitata saranno visualizzati per intero sul display.

Se anzichè <F1> si è premuto <F5> il risultato della ricerca sarà stampato. ATTENZIONE: premendo <F5> in assenza di stampante collegata si provoca il blocco del sistema; per uscirne, premere BREAK .

Se si vuole visualizzare l'intero contenuto del file ADRS.DO nell'ordine in cui i record sono stati introdotti, richiamare ADDRSS, premere <F1> seguito da <ENTER>. Sul display compariranno le prime sei righe del file. Le funzioni F3 e F4 assumeranno rispettivamente la denominazione 'More' (ancora) e 'Quit' (abbandona). Le denominazioni possono essere rimosse dal display premendo <LABEL>. Premendo <F3> si ottiene la visualizzazione delle sei linee di testo successive e la procedura può essere ripetuta sino ad esaurimento del file. Premendo <F4> si interrompe la fase di visualizzazione e si ritorna alla fase precedente (messaggio Adrs:). F3 ed F4 sono disabilitati.

L'esempio che segue illustra quanto sin qui spiegato.

Supponiamo che il contenuto del file ADRS.DO sia quello indicato nel seguito:

1. Carlo Franceschini :(02)987654: via A. Grossich 31 - 20131 Milano

2. Alcide Govi :(010)887711: via Italia 4 - 16010 Grossiglione (GE)

3. Lucia Petitti :(06)123456: str. Aeroporto 198 - 00043 Ciampino (Roma)

4. Tersicore Lancia :(011)5091521: via I. Petitti 1 - 10123 Torino

5. Zeno Rossi :(0125)51234: via Tripoli 93 - 10010 Salerano (TO)

Supponiamo ora di richiamare il programma ADDRSS e di selezionare la funzione 'Find'.

Se la prima stringa di caratteri da ricercare è "Franceschini", l'intero record numero 1. sarà visualizzato sul display.

Se la stringa specificata è "Petitti", saranno visualizzati i record 3 e 4, essendo la stringa presente in entrambi.

Richiediamo ora come stringa di ricerca "Rossi"; ovviamente sarà visualizzato il record 5, ma lo saranno anche i record 1 e 2 in cui la stringa è contenuta nelle parole "Grossich" e "Grossiglione" (si ricordi che non viene fatta distinzione tra lettere minuscole e maiuscole). In questo caso, se si modifica la stringa da ricercare in " rossi " (aggiungendo cioè uno spazio prima e dopo), sarà visualizzato solo il record 5.

Se si vuole evitare l'apparizione di stringhe spurie, è pertanto necessario assegnare alla stringa di selezione almeno un elemento univoco del record da ricercare.

Questa caratteristica può essere evidentemente sfruttata per allargare le prestazioni di ADDRSS. Aggiungendo ai record una sequenza di caratteri inusitata, si possono stabilire delle categorie: il programma ADDRSS potrà poi essere utilizzato per fornire liste per categoria corrispondenti ad ogni sequenza introdotta. Ad esempio, si potrebbero codificare tutti i funzionari di banca con XBB, tutti i responsabili degli acquisti con XAA, tutti i fornitori con XFF, tutti i medici con XDD, etc.

Gli impieghi del programma possono pertanto essere adattati alle esigenze dell'utente. Mediante un uso diversificato di tali codici, e di altri quali il codice di avviamento postale o il prefisso per la teleselezione, è possibile elencare le informazioni di ADRS.DO in molti modi, secondo la categoria, la professione, la localizzazione geografica, etc.

7. IL PROGRAMMA APPLICATIVO SCHEDL

Questo programma fornisce uno strumento per creare, aggiornare e consultare facilmente un'agenda giornaliera, in cui le informazioni possono essere codificate in modi diversi a seconda delle necessità (SCHEDL = agenda elettronica). Come si è già visto per ADDRSS, anche SCHEDL è un programma per la gestione di dati contenuti in un altro file, in questo caso il file NOTE.DO (note = nota, appunto). SCHEDL può quindi essere utilizzato per selezionare e organizzare le informazioni del file NOTE.DO, in modo che funzioni come un'agenda per appuntamenti, un registro delle spese, un diario, un quaderno di appunti in generale, o per altre applicazioni simili.

PREPARAZIONE DEL FILE 'NOTE.DO'

Accedere a TEXT, posizionando il cursore sul nome TEXT del menù principale e premendo <ENTER>. In risposta al messaggio, digitare il nome del file. Per questa particolare applicazione non c'è facoltà di scelta: il nome del file dovrà essere obbligatoriamente NOTE.DO.

Se si richiedesse il programma SCHEDL prima di aver creato il file NOTE.DO, verrebbe visualizzato il messaggio:

NOTE.DO not found	(NOTE.DO non trovato)
Press space bar for menu	(premere barra spazio per MENU)

Le informazioni nel file NOTE.DO sono quelle normalmente presenti in un'agenda per appuntamenti, un diario, ecc. I contenuti sono frutto di una scelta personale. Molti vi scriveranno la lista degli impegni futuri, gli eventi e le date da ricordare, le spese professionali, e così via. Tuttavia, indipendentemente dalla scelta degli argomenti da memorizzare, è consigliabile attenersi alle seguenti regole:

- Le informazioni devono essere organizzate in maniera logica e ordinata, tenendo presente come esse appariranno quando saranno richiamate da SCHEDL.

- Ogni introduzione, o record, deve essere chiusa con <ENTER>. I record dovrebbero avere dimensioni tali da permettere l'introduzione di tutti i dati necessari, e nello stesso tempo contenute, in modo da permettere la visualizzazione su un solo schermo di tutte le informazioni ricercate.

- L'utilizzazione di prefissi per l'identificazione delle informazioni memorizzate sul file, potrà servire a rintracciarle più agevolmente. Le spese, ad esempio, potranno avere il prefisso Lit o $, gli appuntamenti §, gli impegni che richiedono una certa azione ! e così via.

Una volta completato il file NOTE.DO, premere <F8> per chiudere il file e ritornare al menù principale.

UTILIZZO DEL PROGRAMMA SCHEDL

Il programma SCHEDL può ora essere richiamato posizionando il cursore sulla voce SCHEDL e premendo <ENTER>.

Il display assumerà l'aspetto illustrato in Figura 7-1.

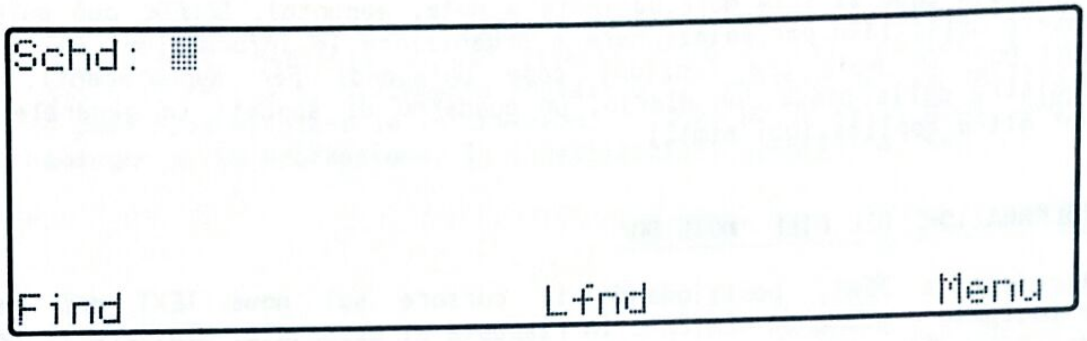

Fig. 7-1 Display ottenuto selezionando SCHEDL

Sulla prima riga appare il messaggio:

Schd: (schedule = registro, orario)

Come si può osservare, in questo programma sono utilizzati solo tre tasti funzione: F1, F5 e F8. Le denominazioni compaiono sul display; per eliminarle è sufficiente premere <LABEL>.

- F1 (Find) ha la stessa funzione che ha nel programma ADDRSS. Premendo <F1> viene visualizzato il messaggio **Find** (trova), dopo il quale occorre digitare la stringa che si vuole ricercare nel file NOTE.DO e premere <ENTER>. Tutti i record contenenti la stringa digitata saranno visualizzati per intero sul display.

- F5 (Lfnd) ha la stessa funzione di F1, con la differenza che il risultato viene inviato in stampa. ATTENZIONE: premendo <F5> in assenza di stampante collegata si provoca il blocco del sistema; per uscirne, premere <BREAK>.

- F8, come nelle altre applicazioni, fa ritornare al menù principale.

Se si vuole visualizzare l'intero contenuto del file NOTE.DO nell'ordine in cui i record sono stati introdotti, richiamare SCHEDL, premere <F1> seguito da <ENTER>. Sul display compariranno le prime sei righe del file. Le funzioni F3 e F4 assumeranno rispettivamente la denominazione 'More' (ancora) e 'Quit' (abbandona). Le denominazioni possono essere rimosse dal display premendo <LABEL>. Premendo <F3> si ottiene la visualizzazione

delle sei linee di testo successive e la procedura può essere ripetuta sino ad esaurimento del file. Premendo <F4> si interrompe la fase di visualizzazione e si ritorna alla fase precedente (messaggio Schd:). F3 ed F4 saranno disabilitati.

L'esempio che segue illustra quanto sin qui spiegato.

Si supponga che le annotazioni sotto riportate rappresentino un estratto del file NOTE.DO.

11/07 $	Spese a Ginevra: Taxi da aereoporto SF 12,70 Colazione SF 25,00 Pranzo SF 67,90 Taxi per aereoporto SF 15,00
12/07 §	10.30 Appuntamento con Z. Rossi
12/07 §	14.30 Incontro con agenzia pubblicità
13/07 §	12.45 Colazione con Prof. Lombardi
14/07 !	BONIFICO 15M per ACS
15/07 *	15.00 Volo per Londra AZ 489 Prenotazione al Grosvenor House tel. 4996363, Tx 24871
16/07 !	Telefonare a Henry a Manchester, ore 10,30

Queste diverse informazioni, per i giorni dall'11 al 16 luglio, possono essere esaminate selettivamente con il programma SCHEDL. In questo esempio particolare i record sono stati codificati come segue:

$ rappresenta le spese professionali

§ rappresenta gli appuntamenti

! rappresenta gli impegni che richiedono una certa azione

* rappresenta i viaggi

Vediamo ora alcuni dei modi in cui le informazioni possono essere selezionate e presentate. Per prima cosa bisogna richiamare il programma SCHEDL, poi premere <F1>.

- Se si sceglie la stringa:

  14/07 !

  saranno presentati tutti gli impegni che richiedono un'azione per quella data (nell'esempio il versamento di un bonifico bancario).

- Analogamente, se la stringa è costituita da uno dei simboli §, $ o *,

si ottiene la lista degli appuntamenti, delle spese o dei viaggi.

- Selezionando invece una data, si ottiene la lista degli impegni di quel particolare giorno.

- Selezionando un nome (ad es. Lombardi) si ottengono gli impegni che coinvolgono questa persona.

Naturalmente l'idea della codifica può essere estesa per soddisfare le esigenze dell'utente. Ad esempio si può usare il doppio codice *$ per i record che concernono sia viaggi che spese, ecc.

Si noti che i simboli usati nell'esempio hanno il solo scopo di dimostrare il principio; l'utente può utilizzare ogni tipo di codifica che sia utile ai suoi fini. In questo contesto, vale la pena di ricordare che la tastiera di M10 è fornita di simboli speciali quali:

- aereoplano <GRPH + SHIFT> + C ,

- telefono <GRPH + SHIFT> + < ,

- automobile <GRPH + SHIFT> + V .

Il file NOTE.DO può essere aggiornato cancellando di tanto in tanto i record sorpassati. In alternativa, se si vogliono mantenere i record concernenti ad esempio le spese o i viaggi, è possibile utilizzare un metodo più sofisticato, consistente nell'aggiungere un carattere al codice, che lo qualifichi come record di data trascorsa. Supponiamo di scegliere la lettera X. Chiedendo allora al programma SCHEDL di cercare la stringa *X si otterrà la lista dei soli viaggi già effettuati.

Le possibilità del sistema sono limitate solo dalle necessità e dall'inventiva dell'utente. Il file NOTE.DO può essere semplice o complesso a seconda di come l'utente lo desidera.

8. IL PROGRAMMA TELCOM

Il programma TELCOM permette il collegamento di M10 con altri computer, sia con trasmissione dei dati su linea telefonica che mediante collegamento diretto. Nel primo caso è necessario utilizzare un modem (modulatore/demodulatore) oppure un accoppiatore acustico con modem incorporato (Modem Coupler) come il modello Olivetti MC10. Per il collegamento diretto, si utilizzerà un cavo specifico, detto cavo 'incrociato'.

Per accedere al programma TELCOM, posizionare il cursore sul nome TELCOM del menù principale e premere <ENTER>. Il display assumerà l'aspetto illustrato in Figura 8-1.

```
37I1E
Telcom:

Find Call Stat Term                    Menu
```

Fig. 8-1 Display ottenuto selezionando TELCOM

Il codice che appare sulla prima linea del display è una lista di parametri di telecomunicazione, trattati con maggior approfondimento più avanti. Sulla seconda riga appare il messaggio **TELCOM:** (programma di telecomunicazione). L'ultima riga visualizza il significato dei tasti funzione quando si accede a TELCOM.

Il programma opera in due modalità differenti: 'Entry' (accesso) e 'Terminal' (terminale). All'inizio esso si pone automaticamente in modalità 'Entry'. Nelle due modalità i tasti funzione F1-F8 hanno significato e utilizzo diverso. La modalità 'Entry' serve essenzialmente per modificare i parametri di comunicazione, mentre la modalità 'Terminal' è usata per lo scambio dati. La modalità 'Terminal' viene richiamata premendo <F4>, dopo aver definito i parametri di comunicazione e aver predisposto M10 ed il modem per la trasmissione.

CONNESSIONE DI M10 A UNA LINEA TELEFONICA

M10 può essere connesso a una linea telefonica tramite modem oppure

accoppiatore acustico con modem incorporato. In quest'ultimo caso, si raccomanda di utilizzare il Modem Coupler Olivetti MC10, appositamente studiato per la trasmissione dati con M10. La descrizione dettagliata di questo dispositivo è fornita in Appendice F.

Nota importante: OGNI PAESE HA UNA LEGISLAZIONE SPECIFICA SULL'USO DI APPARECCHIATURE COLLEGATE ALLA RETE TELEFONICA PUBBLICA. PRIMA DI COLLEGARE M10, E' OPPORTUNO PRENDERE VISIONE DEI REGOLAMENTI VIGENTI PRESSO L'AMMINISTRAZIONE LOCALE PER LE TELECOMUNICAZIONI.

La procedura di connessione di M10 ad MC10 per il collegamento remoto con un elaboratore via telefono è la seguente:

1. Collegare il cavo di MC10 al connettore RS-232C situato sul pannello posteriore di M10.

2. Verificare che il commutatore CAL/ANS, posto sotto ad MC10, sia posizionato su CAL (call = chiamata).

3. Portare l'interruttore ON/OFF di MC10 sulla posizione ON. L'indicatore luminoso emetterà una luce verde lampeggiante.

Il sistema M10 + MC10 è così pronto per essere collegato a un telefono. Se si utilizza un Modem Coupler diverso da MC10, la procedura sarà simile; attenersi in ogni caso alle istruzioni del fornitore per assicurare un collegamento corretto.

MODALITA' ENTRY

Quando si seleziona TELCOM dal menù, si passa sempre in modalità Entry. Questa modalità permette di assegnare ai parametri di comunicazione i valori richiesti; inoltre controlla il passaggio in modalità Terminal e il ritorno al menù principale.

I TASTI FUNZIONE IN MODALITA' ENTRY

In modalità Entry i tasti funzione, illustrati in Figura 8-1, assumono i seguenti significati:

F3 - Stat (stato): elenca i parametri di comunicazione attualmente in vigore e ne consente la modifica; è usato nello scambio di dati con altro computer, e verrà esaminato in dettaglio più avanti in questo stesso capitolo.

F4 - Term (terminale): passa alla modalità Terminale.

F8 - Menù : riporta al menù principale.

I restanti tasti non sono utilizzati.

MODALITA' TERMINAL

Scopo di questa modalità operativa è permettere a M10 di comunicare, via telefono o direttamente, con altri sistemi o con una rete di servizi di informazione, comportandosi come un terminale di elaborazione.

La modalità Terminal viene attivata premendo <F4>, dopo aver richiamato TELCOM da menù principale. Il display assumerà l'aspetto illustrato in Figura 8-2.

```
37I1E
Telcom:Term
▒
Prev Down  Up  Full                    Bye
```

Fig. 8-2 Display in modalità Terminal

I TASTI FUNZIONE IN MODALITA' TERMINAL

Come si può vedere, i tasti funzione F1-F8 controllano ora funzioni diverse rispetto alla modalità Entry. Esse sono:

F1 - Prev (precedente): visualizza le otto righe precedenti di testo; per tornare alla posizione originale, ripremere <F1>.

F2 - Down (carica in giù): scarica sul terminale, memorizzandoli su un file, i dati in arrivo dal sistema ospite, per visualizzarli in un tempo successivo. Premendo <F2> compare il messaggio:

File to Download? (file da scaricare?)

Digitare il nome del file dove si vogliono memorizzare i dati e premere <ENTER>; per arrestare l'operazione, ripremere <F2>.

F3 - Up (carica in sù): invia verso sistema ospite un file di dati preparato in precedenza per la trasmissione. Premendo <F3> compare il messaggio:

File to Upload? (file da inviare?)

Digitare il nome del file da inviare e premere <ENTER>;

compare il messaggio:

Width? (larghezza?)

La risposta a questo messaggio definisce il massimo numero di caratteri di testo consecutivi (compreso tra 10 e 132) che possono essere trasmessi da M10, dopo i quali viene inserito automaticamente - se non previsto - un "ritorno a capo"; digitare il valore scelto e premere <ENTER>.

F4 - Full/Half (duplice/semiduplice): seleziona la modalità di invio dati tra Full Duplex (duplice = trasmissione contemporanea nei due sensi) e Half Duplex (semiduplice = trasmissione alternata nei due sensi). In modalità Full Duplex, richiesta dalla maggior parte dei sistemi ospiti, i caratteri inviati al sistema sono ritrasmessi al terminale per la visualizzazione sul display. In questo modo si può ottenere un controllo visivo sul buon fine dell'invio dati. In Half Duplex invece, i caratteri sono visualizzati mam mano che sono inviati, cosicchè non è possible garantire che ciò che compare sul display sia stato effettivamente ricevuto dal sistema ospite; una linea telefonica scadente, ad esempio, potrebbe aver distorto la trasmissione.

F5 - Echo (eco in stampa): provoca la stampa, sulla stampante collegata a M10, di tutto ciò che viene visualizzato sul display; è usato quando si vuole conservare traccia dello scambio dati con il sistema ospite. F5 agisce come un interruttore: premuto una volta attiva la funzione, ripremuto la disabilita (e toglie la scritta 'Echo' dal display).

F6 ed F7 Non sono utilizzati.

F8 - Bye (addio!): abbandona la modalità Terminal.

I PARAMETRI DI COMUNICAZIONE DATI

In precedenza si è fatto riferimento ai parametri di comunicazione che appaiono sulla prima riga del display quando si richiama TELCOM. Per comunicare con un altro sistema, tali parametri devono essere compatibili con quelli del sistema stesso; si devono quindi selezionare i valori opportuni su M10, entro i limiti riportati in Figura 8-3.

PARAMETRO	VALORI POSSIBILI	SIGNIFICATO
Velocità di modulazione (c)	1 2 3 4 5 6 7 8 9	75 baud 110 baud 300 baud 600 baud 1200 baud 2400 baud 4800 baud 9600 baud 19200 baud
Lunghezza parola (b)	6 7 8	6 bit 7 bit 8 bit
Parità (p)	O E N I	Odd (dispari) Even (pari) No (senza) parità Ignora parità
Bit di stop (s)	1 2	1 bit di stop 2 bit di stop
Stato della linea (x)	E D	Enable (abilita) Disable (disabilita)

Fig. 8-3 Parametri di comunicazione dati (cbpsx)

Nella Figura 8-1 i parametri sono **37I1E**, i valori iniziali di M10. Dalla tabella essi possono essere interpretati come segue:

Velocità di modulazione - 3 valore iniziale pari a 300 baud.

Lunghezza parola - 7 (bit): valore iniziale.

Parità - I (ignora): non effettua controlli di parità

Bit di stop - 1 (bit di stop): valore iniziale

Stato della linea - E (enable): per abilitarla

Non è necessario conoscere il significato preciso dei parametri suddetti per collegarsi in modalità Terminal con un sistema ospite. La loro descrizione in questo paragrafo serve solo a permettere l'adattamento di M10 alle specifiche di collegamento richieste per il sistema stesso.

Per variare i parametri di comunicazione, accedere alla modalità Entry (messaggio **Telcom:**) e premere <F3> (Stat). Sul display appare il messaggio **Stat** ; introdurre il nuovo protocollo di comunicazione (ovvero la stringa di parametri secondo la sequenza richiesta, come in 37I1E) e premere <ENTER>; il messaggio **Telcom:** appare nuovamente. Per verificare che i nuovi valori siano stati registrati da M10, premere nuovamente

<F3>. Dopo il messaggio **Stat**, premere semplicemente <ENTER>: i nuovi valori figurano sul display alla linea successiva. Si ricordi che i parametri di comunicazione sono accessibili e modificabili in modalità Entry, ma che la loro utilizzazione avviene in modalità Terminal.

I valori attuali dei parametri di comunicazione possono essere verificati in qualunque istante, usando <F3> e <ENTER>.

SCAMBIO DATI CON UN SISTEMA OSPITE

La modalità Terminal è stata progettata per consentire il colloquio di M10 con altri elaboratori, sia mediante collegamento diretto che attraverso linea telefonica. In quest'ultimo caso è richiesto l'uso di un modem o di un Modem Coupler. La connessione alla rete telefonica deve essere effettuata prima di accedere alla modalità Terminal.

ACCESSO ALLA MODALITA' TERMINAL

Quando si utilizza il Modem Coupler Olivetti MC10, la procedura è la seguente:

1. Assicurarsi che MC10 sia connesso al connettore RS-232C (sul pannello posteriore di M10), e che il commutatore ANS/CAL sotto MC10 sia posizionato su CAL (call = chiamata).

2. Posizionare su ON l'interruttore ON/OFF di MC10; l'indicatore luminoso emetterà una luce verde lampeggiante.

3. Sollevare il microtelefono e comporre il numero richiesto;
   (se il sistema remoto è a connessione automatica, si udrà una tonalità acuta continua; se è invece a connessione manuale, si dovrà richiedere prima all'operatore di realizzare la connessione).

4. Una volta connessi, premere <F4> per entrare in modalità Terminal.

5. Porre il microtelefono nell'apposito alloggiamento di MC10, inserendo prima il ricevitore e poi il microfono; l'operazione è guidata dalla scritta CORD (cordone), vicino all'alloggiamento del microfono.
   Dopo un momento, l'indicatore luminoso emetterà una luce verde continua.

Quando si passa in modalità Terminal, M10 emette un suono acuto, mentre sull'ultima riga del display compaiono le nuove assegnazioni dei tasti F1-F8 (vedere Figura 8-2). A questo punto, M10 è pronto per stabilire il colloquio (log-on) con il sistema remoto.

Se si utilizza un altro Modem Coupler oppure un collegamento via modem, seguire le istruzioni del fornitore.

LOG-ON SU SISTEMA OSPITE

Quando si ottiene il diritto di accesso ad un elaboratore remoto o ci si abbona ad una rete di servizi di informazione, si ricevono precise istruzioni sulla procedura di registrazione (log-on) con il sistema ospite. Le procedure di log-on per i computer variano soltanto nei particolari, mentre le caratteristiche di base - come l'impostazione dell'identificativo di utente e della parola chiave - sono comuni a tutte. Il modo più semplice per chiarire una tipica procedura di log-on è quello di utilizzare un esempio.

Supponiamo di essere utenti di un servizio informativo computerizzato, la Banca Informazioni sul Credito Industriale (BICI), il cui numero telefonico è (06) 987654. Supponiamo ancora che BICI abbia assegnato all'utente l'identificativo (ID) 5138Q e la parola chiave (password) 'baronerosso', insieme alle seguenti istruzioni per il log-on:

a) Comunicare al sistema la predisposizione al log-on mediante invio di Control-C.

b) Attendere il messaggio **ID utente:** ed introdurre il numero di identificazione assegnato.

c) Attendere il messaggio **Password:** ed introdurre la parola chiave.

d) Attendere almeno 3 secondi, quindi inviare l'interrogazione OK?.

e) Il sistema invierà allora il messaggio **Log-on completato!**.

La procedura per il log-on è la seguente:

1. Collegare il Modem Coupler Olivetti MC10 al connettore RS-232C sul pannello posteriore di M10, posizionare il commutatore ANS/CAL su CAL e portare l'interruttore ON/OFF su ON, come già precedentemente spiegato. (Nel caso si utilizzi un Modem Coupler di altra marca, seguire attentamente le istruzioni del costruttore per assicurarsi che esso sia predisposto per la chiamata.)

2. Accedere al programma TELCOM dal menù principale e verificare che i parametri di comunicazione visualizzati sulla prima riga del display corrispondano alle specifiche BICI. In caso contrario, premere <F3> e introdurre i valori appropriati, ricordando che con MC10 la massima velocità di modulazione è di 300 baud.

3. Sollevare il microtelefono e comporre il numero di BICI, (06) 987654. Non appena stabilita la comunicazione, porre il microtelefono nell'apposito alloggiamento di MC10, come descritto nel paragrafo su "Accesso alla modalità Terminal".

4. Premere <F4> (Term), per accedere alla modalità Terminal. M10 emetterà una tonalità acuta per confermare, e contemporaneamente appariranno sul display i nuovi significati dei tasti F1-F8.

*** Seguire ora le istruzioni BICI per il log-on. ***

5. Lasciare trascorrere qualche secondo, quindi premere <CTRL> + C per indicare al sistema BICI la disponibilità al log-on.

6. Attendere il messaggio **ID utente:** sul display, quindi digitare l'identificativo 5138Q e premere <ENTER>.

7. In risposta al messaggio **Password:** che apparirà successivamente, digitare la parola chiave 'baronerosso' e premere <ENTER>. La parolachiave non compare sul display, per proteggerne la segretezza, ma viene comunque trasmessa.

8. Dopo almeno tre secondi, digitare la domanda OK?. Se tutto è avvenuto regolarmente, comparirà il messaggio **Log-on completato!**, ed M10 è pronto per lo scambio dati il sistema BICI.

USO DELLE PRESTAZIONI DI INVIO E SCARICAMENTO FILE

Nella precedente descrizione della modalità Terminal, si è fatto cenno a queste due prestazioni a proposito del significato dei tasti funzione. Per poterle utilizzare con sicurezza, occorre tuttavia descrivere la procedura con maggiori dettagli.

Prestazione di invio (Upload)

Permette di trasmettere a un altro computer un file di informazioni preparato in precedenza. La procedura è la seguente:

1. Seguendo le istruzioni contenute nel Capitolo 5, preparare un file TEXT contenente le informazioni da trasferire, e memorizzarlo.

2. Stabilita la comunicazione con il sistema remoto, per mezzo del programma TELCOM, ed effettuata le predisposizione allo scambio dati, premere <F3> (Up). Sul display compare il messaggio:

   File to upload? (file da inviare?)

3. Digitare il nome del file preparato in precedenza, e premere <ENTER>.

4. Compare allora il messaggio:

   Width? (larghezza?)

   Digitare il valore, compreso tra 10 e 132, e premere <ENTER>.

Nota: Il valore definito con Width? determina la larghezza di riga del file da inviare. La larghezza 'fisica' di riga su M10 è di 40 caratteri (la larghezza del display). Tuttavia, poichè la caratteristica di spostamento automatico a capo delle parole (presente in TEXT) rende superflui i ritorni a capo nell'ambito di un paragrafo, una riga di testo (riga 'logica') può essere molto lunga. Alcuni sistemi di comunicazione impongono dei limiti alla lunghezza delle righe trasmesse. Tale limitazione può essere risolta dalla risposta al messaggio **Width?**: se M10 non trova un ritorno a capo entro il numero di caratteri fissato, ne introduce forzatamente uno. Tutti i ritorni a capo presenti nel file

vengono comunque trasmessi. Premendo soltanto <ENTER> non verrà aggiunto alcun ritorno a capo.

Prestazione di scaricamento (Download)

Permette di memorizzare in un file (scaricare) i dati trasmessi da un altro computer, per utilizzo differito. Ciò è particolarmente utile quando i dati sono inviati in blocchi lunghi e richiedono una interpretazione o elaborazione successiva. La procedura è la seguente:

1. Supponiamo di avere già attivato la comunicazione; M10 si troverà in TELCOM e in modalità Terminal. Impostare i comandi richiesti dal sistema ospite per l'invio dei dati, MA NON PREMERE ANCORA <ENTER>.

   Premere <F2> (Down). Sul display compare il messaggio:

   File to download? (file da scaricare?)

2. Introdurre il nome del file dove le informazioni in arrivo devono essere memorizzate e premere <ENTER>. Non è necessario creare il file in precedenza. Scegliere un nome non più lungo di sei caratteri che rispetti le regole dei file TEXT; esso verrà creato e memorizzato automaticamente.

3. Premere <ENTER> per completare il comando al sistema ospite e avviare la procedura di scaricamento.

4. Durante la trasmissione, la scritta 'Down' sul display appare "in negativo", a conferma che i dati in arrivo sono memorizzati.

5. A fine trasmissione, chiudere la procedura di scaricamento premendo nuovamente <F2>.

6. Per terminare la comunicazione, premere <F8> (Bye).

Per accedere ai dati ricevuti, tornare al menù principale premendo <F8> (Menu) e selezionare il file in cui sono stati scaricati. Da questo punto si può operare come su un qualsiasi file TEXT. Se il sistema ospite ha inserito dei caratteri di controllo nelle informazioni trasmesse, potrà essere richiesta qualche operazione di editing.

Se il file scaricato è un programma BASIC, si deve entrare in BASIC e caricare il file. Questa operazione lo converte in formato binario per l'esecuzione o per memorizzarlo come file con suffisso .BA. Una volta memorizzato il programma come file BASIC, si può cancellare la versione con suffisso .DO.

9. UTILIZZO DEL REGISTRATORE DI NASTRO A CASSETTA

Si è già detto in diverse occasioni che i file creati su M10 possono venire archiviati su cassette di nastro magnetico. Questa operazione è molto utile per ottenere copie di riserva dei file o per far spazio in memoria, quando vi si devono registrare nuove informazioni.

I programmi applicativi BASIC e TEXT forniscono i comandi necessari per registrare un file su nastro e per caricare un file da nastro. Sotto quest'aspetto, l'uso del registratore può considerarsi una prestazione standard di M10; questo capitolo lo descrive in dettaglio.

COLLEGAMENTO DI M10 A UN REGISTRATORE A CASSETTA.

Collegando M10 a un registratore di nastro a cassetta si possono eseguire sia operazioni di registrazione di file da M10 a nastro (Save = salvare), che operazioni di lettura di file da nastro a M10 (Load = caricare). Prima di effettuare il collegamento, inserire la cassetta e riavvolgerla sino al punto dove inizierà la registrazione, usando il contatore per raggiungere la posizione voluta. Si consiglia di annotare sempre la lettura del contatore quando si registrano i file su nastro; ciò semplifica il compito di ritrovarli. Si può usare qualsiasi tipo di registratore purchè, in aggiunta alle normali funzioni ('RECORD', 'PLAY', 'STOP', 'REWIND' etc.), abbia anche le seguenti caratteristiche:

- Jack (presa) di ingresso per microfono, MIC
- Jack (presa) di uscita per auricolare, EAR
- Jack (presa) di controllo remoto, REM
- Contatore di nastro

Il collegamento fra M10 e registratore a cassetta è illustrato in Figura 9-1. Il cavo di collegamento può essere fornito dal concessionario o rivenditore Olivetti. Questo cavo ha da un lato tre spinette per collegarsi alle prese del registratore a cassetta, e dall'altro un connettore rotondo a otto piedini per il collegamento a M10. Le tre spinette sono di colore rosso, bianco e nero. La spinetta nera ha un piedino di connessione più piccolo delle altre due.

1. Collegare il connettore rotondo a otto piedini, posto a un'estremità del cavo, all'ingresso TAPE sul pannello posteriore di M10.
2. Collegare la spinetta bianca al jack di uscita, EAR.
3. Collegare la spinetta nera al jack di controllo remoto, REM.
4. Collegare la spinetta rossa al jack di ingresso, MIC.

5. Accendere il registratore a cassette ed M10

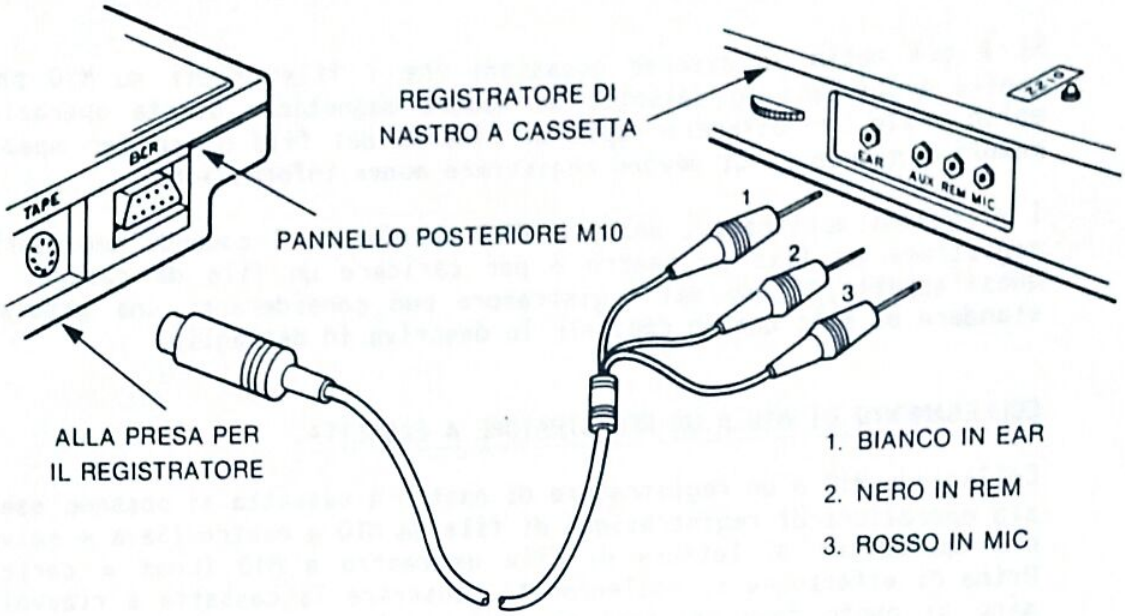

Fig. 9-1 Collegamento di M10 a un registratore a cassetta

Cosìcollegato al registratore, M10 può inviare due segnali di controllo:

MOTOR ON - (motore abilitato): pone il registratore in controllo locale; il suo funzionamento è comandato dai tasti.

MOTOR OFF - (motore disabilitato): inibisce l'azione dei tasti, assegnandone il controllo a M10; il motore si avvia e si arresta automaticamente quando si trasferiscono dei file.

Questi sono entrambi comandi BASIC ; il loro uso è estremamente semplice, anche per chi non abbia mai utilizzato un linguaggio di programmazione:

1. Posizionare il cursore su BASIC (menù principale), premere <ENTER>.

2. Il display si configura come in Figura 4-1.

3. Digitare il comando (immediato) MOTOR OFF (oppure MOTOR ON), premere <ENTER>

Prima di un'operazione di 'Load' o 'Save' su cassetta, è necessario inviare il comando MOTOR OFF. Per riavvolgere o posizionare manualmente il nastro, senza togliere la spinetta nera, inviare il comando MOTOR ON.

REGISTRAZIONE DI UN FILE SU NASTRO A CASSETTA

Quest'operazione può essere eseguita utilizzando i tasti funzione di TEXT, per i file contraddistinti dal suffisso .DO, ed i tasti funzione di BASIC per i file con suffisso .BA. In ogni caso occorre posizionare in precedenza il nastro nel punto in cui si vuole registrare il file. Dopo aver effettuato il collegamento ed inviato il comando MOTOR OFF, come descritto nel paragrafo precedente, premere contemporaneamente i tasti RECORD e PLAY (registrazione).

REGISTRAZIONE DI UN FILE TEXT

1. Accedere al file TEXT, posizionando opportunamente il cursore sul menù principale e premendo <ENTER>.

2. Premere <F3> (Save); compare il messaggio:

   Save to: (salvare verso:)

3. Scegliere un nome per il file di destinazione (su nastro), su cui verrà registrato il file da salvare. Tale nome deve sottostare alle regole di nomenclatura dei file (vedere Capitolo 5).

   Digitare il nome scelto e premere <ENTER>.

4. Il registratore parte automaticamente e si arresta a fine operazione; scompare anche il messaggio sul display.

Ora il file è registrato (salvato) su nastro con il nome prescelto.

REGISTRAZIONE DI UN FILE BASIC

La procedura per i file BASIC è lievemente diversa.

1. Accedere alla modalità BASIC da menù principale.

2. Premere il tasto <F2> (Load) e, in risposta al messaggio **Load "**, digitare il nome del file (in memoria) che si vuole salvare su nastro; quindi premere <ENTER>.

3. Premere ora <F3> (Save) e, in risposta al messaggio **Save "** che viene visualizzato, impostare il comando:

   CAS: nomefile-destinazione <ENTER>

   dove 'nomefile-destinazione' è il nome scelto per il file di destinazione (su nastro), su cui verrà registrato il file da salvare. Tale nome deve sottostare alle regole di nomenclatura dei file.

4. Il registratore parte automaticamente e si arresta a fine operazione; scompare anche il messaggio sul display.

Ora il file è registrato (salvato) su nastro con il nome prescelto.

CARICAMENTO DI UN FILE DA NASTRO A CASSETTA

Anche qui la procedura è lievemente diversa a seconda che si tratti di file in ambiente TEXT o BASIC.

CARICAMENTO DI UN FILE IN MODALITA' TEXT

1. Riavvolgere il nastro fino a che il contatore indichi la posizione corrispondente all'inizio del file da caricare su M10. Se la posizione non è nota, posizionare il nastro all'inizio; M10 scorrerà l'intero nastro per cercare il file richiesto.

2. Collegare il registratore a M10 come descritto in precedenza e premere il tasto PLAY (ascolto). Il registratore (che ha ricevuto il comando MOTOR OFF) non parte fino a quando non è attivato da M10.

3. Si può creare un file TEXT nuovo ove memorizzare il file proveniente da cassetta oppure scegliere un file TEXT già esistente; in quest'ultimo caso, il testo registrato sul nastro viene accodato al testo già presente in memoria.

   Accedere al file TEXT, secondo le procedure note.

4. Premere il tasto funzione <F2>. Apparirà il messaggio:

   Load from: (carica da:)

   Digitare il nome del file di provenienza (su nastro), con il quale era stata eseguita a sua tempo la registrazione, e premere <ENTER>.

5. M10 inizierà ora la ricerca sulla cassetta del file richiesto, emettendo una tonalità acuta quando incontra dati registrati. Ogni volta che si incontra un file diverso da quello cercato, compare il messaggio:

   SKIP: nomefile-su-nastro (salto il file:....)

   dove 'nomefile-su-nastro' è il nome del file incontrato.

   Una volta localizzato il file corretto, M10 visualizza:

   FOUND: nomefile-provenienza (trovato il file:....)

   dove 'nomefile-provenienza' è il nome del file richiesto.

Quando appare questo messaggio, il file su nastro viene caricato nel file TEXT prescelto. In un'operazione di 'Load' il file su nastro non sostituisce mai il file TEXT (se ne esiste uno), ma viene 'appeso' alla fine di questi.

CARICAMENTO DI UN FILE BASIC

1. Riavvolgere il nastro a cassetta nella posizione alla quale inizia il file da caricare, collegare il registratore e premere il tasto PLAY, come descritto nei punti 1 e 2 del paragrafo precedente.

2. Da menù principale, attivare il programma BASIC e premere <F2>. Viene visualizzato il messaggio:

   Load "

3. Digitare il seguente comando:

   CAS: nomefile-provenienza <ENTER>

   dove 'nomefile-provenienza' è quello con il quale il file è stato originalmente registrato su nastro.

4. M10 inizierà ora la ricerca sulla cassetta del file richiesto, emettendo una tonalità acuta quando incontra dati registrati. Ogni volta che si incontra un file diverso da quello cercato, compare il messaggio:

   SKIP: nomefile-su-nastro (salto il file:....)

   dove 'nomefile-su-nastro' è il nome del file incontrato.

   Una volta localizzato il file corretto, M10 visualizza:

   FOUND: nomefile-provenienza (trovato il file:....)

   dove 'nomefile-provenienza' è il nome del file richiesto.

Quando appare questo messaggio, il file su nastro viene caricato nel file BASIC prescelto.

Si noti che nelle operazioni di 'Load' e di 'Save' in ambiente BASIC, è necessario, in risposta ai messaggi, specificare non solo il nome del file ma anche quello dell'unità periferica, in questo caso CAS. Questo perchè il programma BASIC può anche accedere ad altre periferiche, ed inoltre salvare o caricare file in memoria. Le operazioni di 'Load' e di 'Save', in ambiente BASIC, possono anche essere effettuate da comandi diretti, come i comandi CLOAD e CSAVE. Ulteriori informazioni si possono trovare nella parte II di questo manuale.

MANUTENZIONE

Nell'utilizzo di un registratore a cassette di nastro magnetico in collegamento a computer, e pertanto anche in questo caso, è importante osservare scrupolosamente le raccomandazioni del fabbricante sulla manutenzione e la pulizia. In particolare, assicurarsi che le testine siano ben pulite e smagnetizzate, per evitare danni o "rumori" nei file registrati.

PARTE II

LINGUAGGIO BASIC: GUIDA DI RIFERIMENTO

10. INTRODUZIONE AL BASIC

I prossimi capitoli sono interamente dedicati al BASIC (Beginners All-purpose Symbolic Instruction Code), che è il linguaggio di programmazione ad alto livello di M10. Esistono diverse versioni di BASIC, ma tutte le descrizioni o riferimenti di questo manuale si applicano alla sola versione M10. Nel Capitolo 4, è stata già fornita una breve descrizione del 'programma applicativo' BASIC, sicchè certi aspetti del BASIC M10 sono già noti: come si accede all'ambiente BASIC, le tre modalità operative, il concetto di linea, il significato dei tasti funzione etc. I capitoli che seguono trattano della scrittura ed esecuzione dei programmi, dei dati e degli operatori aritmetici, relazionali e logici; la parte finale riporta la descrizione in ordine alfabetico dei comandi BASIC.

NOTAZIONE: La convenzione riportata nel Capitolo 2 rimane valida anche in questa parte del manuale. Per quanto riguarda la sintassi del BASIC, devono essere aggiunti anche i seguenti simboli, che devono essere introdotti così come sono.

Virgolette	"	Punto	.
Virgola	,	Due punti	:
Punto e virgola	;	Simbolo di addizione	+
Simbolo di sottrazione	-	Simbolo di moltiplicazione	*
Simbolo di divisione	/	Simbolo di esponentê	^
Barra all'indietro (°)	\	Uguale	=
Maggiore di	>	Minore di	<
Punto interrogativo	?	Simbolo di percentuale	%
Punto esclamativo	!	Simbolo di numero (°)	#
Simbolo di dollaro	$	Parentesi	()

(°) sulla tastiera italiana il simbolo 'barra all'indietro' è sostituito dal simbolo 'c cediglia' (ç) e il 'simbolo di numero' dal simbolo di lira sterlina (£).

11. PROGRAMMI BASIC

ORGANIZZAZIONE

In M10, i programmi BASIC sono residenti in memoria. Sul menù principale possono essere visualizzati sino a 19 nomi di file (programmi e dati); ciò determina il massimo numero di programmi (e di file dati) che possono risiedere contemporaneamente in memoria. Come visto nel Capitolo 9, i programmi (e i file dati) possono essere archiviati su cassetta di nastro magnetico; questo permette di allargare notevolmente la capacità di memorizzazione di M10. Quando si registrano programmi e file dati su cassetta, è buona norma fare sempre due copie (su due cassette diverse): una per l'uso normale e una di riserva (back-up) in caso di danneggiamento del file o dell'intera cassetta.

DOCUMENTAZIONE DI UN PROGRAMMA

E' molto importante che i programmi siano costruiti in modo logico e che siano ben documentati. Una procedura può sembrare ovvia quando la si scrive, ma può risultare incomprensibile dopo qualche tempo, a meno che non sia stata documentata in modo completo. Ciò si può effettuare sia con l'istruzione REM che inserendo un campo commento, cioè una stringa di caratteri preceduta da un apostrofo (') e chiusa da <ENTER>.

REM

I commenti possono occupare una linea intera o far parte di una linea in cui vi sono altre istruzioni. In quest'ultimo caso, i commenti devono essere inseriti a fine linea. Sia le istruzioni REM, sia i campi commento, possono essere inseriti in qualunque punto del programma; non sono istruzioni eseguibili, ma appaiono nei listing.

CREAZIONE DI UN PROGRAMMA

Un programma BASIC viene creato su M10: digitando le istruzioni in tastiera, verificando tramite la lista (listing) che esse siano corrette, memorizzando poi il programma su nastro per poterlo riutilizzare in tempi successivi.

INTRODUZIONE DI UN PROGRAMMA

Ogni riga di programma deve iniziare con un numero di riga, che è introdotto da tastiera.

Le righe di programma possono essere introdotte in ordine qualsiasi; il sistema provvederà a riorganizzarle in ordine numerico durante il

listing. Quest'ultimo può essere eseguito in ogni momento, con le modalità descritte più avanti. Il programma può anche essere memorizzato ad ogni fase dello sviluppo, con le modalità che verranno fornite. Tutti i comandi, le istruzioni e le funzioni che si possono usare su M10, sono descritte in dettaglio nel Capitolo 15.

LISTA DI UN PROGRAMMA

Per eseguire una lista (listing) del programma su display è disponibile il comando LIST. Se il listing deve essere eseguito sulla stampante, si utilizzerà il comando LLIST.

Il comando LIST può essere usato per visualizzare su display una o più righe di programma come segue:

LIST n - m <ENTER>

dove n è la prima linea da visualizzare e m l'ultima. Se non si specifica n vengono visualizzate tutte le righe dall'inizio fino ad m; se non si specifica m vengono visualizzate tutte le righe da n alla fine; se si specifica solo n (senza il trattino) viene visualizzata solo la riga n. LIST <ENTER> lista l'intero programma. LIST . visualizza l'ultima riga utilizzata dal programma BASIC.

Un programma di cui si è fatto il listing viene registrato in memoria, ma la sua sintassi non è ancora stata verificata.

Il comando LLIST opera esattamente nello stesso modo, ma il listing viene eseguito su stampante anzichè su display.

Per la descrizione completa di questi due comandi, vedere il Capitolo 15.

REGISTRAZIONE DI UN PROGRAMMA

Un programma contenuto in memoria può essere registrato su cassetta di nastro magnetico, o su altro dispositivo collegato ad M10, tramite il comando SAVE (salva, registra). Questi ha infatti alcune opzioni, disponibili per unità periferiche diverse, e che sono descritte nel Capitolo 15.

Per registrare il programma attuale su cassetta si deve usare il comando 'CSAVE nomefile-destinazione', dove nomefile-destinazione (massimo sei caratteri) identifica il nome del file su nastro che conterrà il programma. E' possibile verificare che il programma sia stato trasferito correttamente, utilizzando il comando 'CLOAD? nomefile-destinazione' che confronta il file su nastro con quello in memoria. Se c'è qualche discordanza viene visualizzato il messaggio **Verify failed** (verifica non esatta), mentre in caso contrario appare il messaggio **Ok**. Per arrestare una registrazione in corso, premere <BREAK>. Per la descrizione completa dei comandi CSAVE e CLOAD? vedere il Capitolo 15.

USO DI UN PROGRAMMA

Per usare un programma è necessario prima caricarlo in memoria, quindi eseguirlo. Se viene eseguito con successo, saranno visualizzati o stampati i risultati. Se vi sono problemi che ne impediscono l'esecuzione, ad esempio errori sintattici, questi saranno segnalati dal sistema. La procedura di correzione è descritta più avanti.

CARICAMENTO DI UN PROGRAMMA

Il comando LOAD (carica) ha diverse opzioni che permettono il caricamento in memoria di programmi registrati su periferiche, e che sono descritte in dettaglio nel Capitolo 15. Nel caso di programmi registrati su cassetta di nastro magnetico tramite CSAVE, il caricamento in memoria avverrà tramite il comando 'CLOAD nomefile-provenienza', dove nomefile-provenienza è il nome del file che contiene il programma richiesto. Se il nome è omesso, viene caricato in memoria il primo programma trovato sul nastro. Per arrestare un'operazione di caricamento, premere <BREAK>.

E' possibile caricare da cassetta ed eseguire immediatamente un programma con un solo comando, come viene descritto più avanti.

ESECUZIONE DI UN PROGRAMMA

Per eseguire un programma si usa il comando RUN (esegui). Digitando il comando:

RUN <ENTER>

il sistema tenterà di eseguire il programma caricato per ultimo in memoria. Se il programma da eseguire non è stato ancora caricato, si può effettuare il caricamento, e quindi l'esecuzione, usando:

RUN "CAS:nomefile-provenienza" <ENTER>

dove nomefile-provenienza è il nome del file che contiene il programma richiesto.

Se vi sono errori che ne impediscono l'esecuzione, questi saranno segnalati dal sistema. La procedura di correzione è descritta più avanti.

Se il programma richiede la digitazione di dati da parte dell'utente, viene visualizzato il carattere ?. Normalmente questo carattere è preceduto da un messaggio che spiega il tipo di dato richiesto (fare riferimento all'istruzione INPUT nel Capitolo 15, per la spiegazione di come ciò deve essere previsto durante la stesura di un programma). Digitare il dato richiesto e inviarlo al sistema mediante il tasto <ENTER>.

STAMPA

I risultati di un programma possono essere visualizzati mediante l'istruzione PRINT (stampa), oppure inviati alla stampante mediante l'istruzione LPRINT.

E' possibile stampare i dati con un formato specifico, definendolo per mezzo delle istruzioni PRINT USING o LPRINT USING. Quest'ultima è particolarmente utile quando si vogliano stampare file o risultati di programmi in un formato diverso da quello col quale essi vengono visualizzati. Se si usa soltanto l'istruzione LPRINT si otterrà in stampa un formato identico a quello del display.

Per inviare i dati ad una stampante collegata all'interfaccia seriale RS-232C, occorre che essi siano prima scritti in un file: ciò è spiegato nel Capitolo 15.

CORREZIONE DEGLI ERRORI IN UN PROGRAMMA

Gli errori commessi nella scrittura di un programma, ad esempio errori sintattici, vengono scoperti dal sistema quando se ne tenta l'esecuzione per la prima volta. La segnalazione di errore è accompagnata dal numero di riga in cui è stato identificato, il che ne facilita la correzione.

ERRORI

Una lista completa dei codici di errore è riportata in Appendice B. Ogni codice è composto da un nome di due caratteri, che viene visualizzato, e da un numero. Quest'ultimo è un numero intero compreso fra 1 a 255; molti numeri non sono utilizzati, e sono disponibili per ampliamenti del sistema BASIC o per utilizzazione da parte del programmatore. Oltre agli errori che si possono rilevare nella fase di scrittura di un programma, ve ne sono altri che si possono presentare in fasi diverse, come ad esempio errori di IO (Input/Output), che indicano una situazione anomala presentatasi durante un'operazione di scrittura o di lettura di dati.

L'utente può definire un errore utilizzando l'istruzione ERROR n, in cui n è un intero che specifica il codice di errore. Per chiarimenti su quest'istruzione consultare il Capitolo 15.

RICERCA DEGLI ERRORI

Si possono inserire in un programma delle routine per il trattamento degli errori, utilizzando l'istruzione ON ERROR GOTO n per indicare la prima riga, n, di tali routine. In qualunque istante dopo un'istruzione ON ERROR GOTO, un errore provocherà un salto alla routine. La funzione di ricerca viene soppressa dopo l'istruzione ON ERROR GOTO 0.

MODIFICHE A UN PROGRAMMA

Un programma può essere modificato solo quando è residente in memoria. Se non lo è, é necessario caricarvelo, come descritto in precedenza.

programma, può essere più semplice usare la prestazione di 'editing'. Il comando EDIT porta il sistema in modalità TEXT, in cui è abilitata questa prestazione. La descrizione completa di questo comando è fornita nel Capitolo 15.

La modalità TEXT, descritta nel Capitolo 5, viene utilizzata per creare o modificare i file. Essa dispone di un certo numero di comandi che semplificano la creazione e la modifica dei file e che, ad eccezione della prestazione di riposizionamento a capo automatico delle parole, sono utilizzabili anche per la modifica dei programmi.

FUSIONE DI DUE PROGRAMMI

Un programma su cassetta può essere fuso con uno residente in memoria mediante l'uso del comando:

```
MERGE "CAS:nomefile"
```

In questo modo le righe del programma su cassetta sono inserite nel programma in memoria secondo l'ordinamento numerico. Se una riga del programma su cassetta ha lo stesso numero di una del programma in memoria, quest'ultima viene sostituita, ed il suo contenuto iniziale è perso.

12. DATI E ARITMETICA

I DATI NEI PROGRAMMI BASIC

I dati che un programma BASIC può prendere in considerazione possono essere delle costanti o delle variabili. Una costante ha il medesimo valore per tutta l'esecuzione del programma, mentre il valore di una variabile può cambiare. Alle variabili sono quindi associati dei nomi, utilizzati come riferimenti nella scrittura del programma; ad esempio la formula

C + 273

fornisce la temperatura in gradi Kelvin a partire dalla variabile C, che rappresenta la temperatura espressa in gradi Celsius. Un programma che utilizzi quest'espressione può essere eseguito diverse volte con valori di C sempre diversi.

I nomi delle variabili possono anche essere più lunghi di due caratteri, ma solo i primi due vengono presi in considerazione; devono pertanto essere univoci. Si possono assegnare nomi anche alle costanti, per comodità: ad esempio è più semplice digitare un nome breve piuttosto che una costante molto lunga. Una parola riservata non può essere usata come nome o parte di un nome. Si raccomanda quindi di limitare l'estensione dei nomi a uno o due caratteri, in modo tale che le parole riservate da evitare siano solamente IF, ON, OR e TO.

'Espressione' è un nome generico per un elemento, o una combinazione di elementi, di dati. Può contenere delle costanti, delle variabili o entrambe, ma non può contenere insieme dati numerici e stringhe di caratteri. La formula citata è un'espressione numerica, ed è necessario assegnare un valore alla variabile numerica C prima di poter calcolare l'espressione stessa.

DATI STRINGA

Le stringhe (sequenze di caratteri ISO) sono utilizzate per memorizzare dati non numerici, come nomi, indirizzi, codici ecc. e possono avere lunghezza variabile fra 0 e 255 caratteri. Il BASIC assegna loro spazio in memoria secondo necessità. Una stringa con lunghezza zero è detta stringa "nulla", ed è rappresentata da una coppia di virgolette ("").

DATI NUMERICI

Il BASIC può trattare dati numerici di tre tipi differenti a seconda della precisione richiesta nei calcoli, e cioè numeri interi, numeri in semplice precisione e numeri in doppia precisione. Tutti i dati numerici sono introdotti in formato decimale (cioè in base 10).

Numeri interi

Permettono la maggiore efficienza, sia in termini di occupazione di memoria che di velocità di calcolo. Il campo di definizione è però limitato ai numeri interi compresi tra -32768 e 32767.

Numeri in semplice precisione

Rappresentazione di uso generale nelle applicazioni BASIC. Il campo di definizione è compreso fra $\pm 10^{-64}$ e $\pm 10^{62}$. Nei calcoli sono usate fino a sette cifre significative, mentre ne vengono visualizzate o stampate fino a sei, con arrotondamento per eccesso (se la settima è maggiore o uguale a 5). Per memorizzare numeri in semplice precisione è necessaria più memoria, e i tempi di calcolo sono più lunghi che per i numeri interi.

Numeri in doppia precisione

Consente la massima precisione nella rappresentazione dati in BASIC. Il campo di definizione è compreso fra $\pm 10^{-64}$ e $\pm 10^{62}$. Nei calcoli sono usate fino a sedici cifre significative, mentre ne vengono visualizzate o stampate fino a quindici, con arrotondamento per eccesso (se la sedicesima è maggiore o uguale a 5). Questa rappresentazione richiede ancora più memoria e tempi di calcolo ancora più lunghi.

CLASSIFICAZIONE DEI DATI

Solitamente una stringa è riconoscibile in quanto racchiusa fra virgolette. Tuttavia, durante l'introduzione dei dati o in un'istruzione DATA ciò non è normalmente richiesto.

Le costanti numeriche sono classificate nella categoria di minima precisione possibile, cioè:

- se un numero ha più di sette cifre significative, esso è un numero in doppia precisione
- se un numero è senza decimali, ed è compreso nell'intervallo fra -32768 e 32767, esso è un numero intero
- se un numero non appartiene a nessuna di queste due categorie, esso è un numero in semplice precisione.

Le variabili numeriche sono classificate inizialmente come numeri in doppia precisione.

MODIFICA DELLA CLASSIFICAZIONE

Sia le costanti che le variabili possono essere assegnate a un tipo diverso da quello definito automaticamente. Per le costanti, l'assegnazione del tipo può essere specificata solamente aggiungendo alla fine un carattere di dichiarazione, come indicato in Figura 12-1.

Le variabili possono essere assegnate a tipi diversi all'inizio del programma, utilizzando una delle seguenti istruzioni:

- DEFINT ident - specifica variabili intere,
- DEFSNG ident - specifica variabili in semplice precisione,
- DEFDBL ident - specifica variabili in doppia precisione,
- DEFSTR ident - specifica variabili stringa;

dove ident definisce le (lettere) iniziali che identificano l'appartenenza di una variabile a un certo tipo. Ad esempio:

```
10 DEFINT A-C, M, X-Z
```

definisce tutte le variabili del programma, il cui nome inizia per A, B, C, M, X, Y, Z, come variabili intere. Il formato delle istruzioni DEF è riportato in dettaglio nel Capitolo 15.

E' possibile superare queste assegnazioni di carattere generale, utilizzando in casi specifici i caratteri di assegnazione elencati in Figura 12-1, come per le costanti.

CARATTERE	SIGNIFICATO
%	Intero
!	Semplice precisione
£	Doppia precisione
$	Stringa

Fig. 12-1 Caratteri di assegnazione del tipo

A meno che le variabili e le costanti numeriche non siano definite come numeri interi o in semplice precisione per mezzo delle istruzioni DEF o dei caratteri di assegnazione, tutti i calcoli sono eseguiti in doppia precisione.

CONVERSIONI NUMERICHE

Talvolta è necessario assegnare una costante di un certo tipo a una variabile di tipo diverso o convertire un tipo di variabile in un altro. Ciò può essere effettuato usando la forma di assegnazione a = b di cui viene riportato qualche esempio nel seguito.

Conversioni da semplice o doppia precisione ad intero e da doppia a semplice precisione implicano troncamenti o arrotondamenti, e possono produrre errori di overflow (superamento) nel caso che la variabile da convertire non sia contenuta nell'intervallo tipico del tipo in cui essa

viene convertita. Conversioni di variabili intere non producono errori.

Esempio 1

Conversione della variabile in semplice precisione A (=23.56), in variabile intera.
Seguire la sequenza di comandi BASIC:

```
A% = 23.56 <ENTER>
Ok
?A% <ENTER>
 23
Ok
```

Si noti che il numero è stato troncato ma non arrotondato.

Esempio 2

Conversione della variabile in doppia precisione A (=5.0069999999), in variabile in semplice precisione.
Seguire la sequenza di comandi BASIC:

```
A! = 5.0069999999 <ENTER>
Ok
?A! <ENTER>
 5.007
Ok
```

In questi esempi le linee contenenti <ENTER> sono comandi da introdurre, le altre righe sono fornite dal programma BASIC.

MATRICI

Una 'matrice' (array) è un insieme di variabili dello stesso tipo, aventi tutte lo stesso nome ma distinguibili tramite il valore di uno o più indici indicati tra parentesi dopo il nome; ad esempio:

A(0), A(1), A(2)

sono i tre elementi di una matrice a una dimensione, A, che ha 2 come indice massimo. Questo viene detto l'estremo superiore della dimensione.

Una matrice può avere un numero di dimensioni qualsiasi purchè vi sia sufficiente spazio in memoria per contenerla. Analogamente, ogni dimensione può avere qualunque numero di elementi, purchè la memoria sia sufficiente. Il BASIC assegna spazio in memoria alla matrice non appena questa viene specificata.

Quelli che seguono sono gli elementi di una matrice a due dimensioni, in cui gli indici massimi sono rispettivamente n e m.

B(0,0)	B(0,1)	B(0,2)	- - - - - -	B(0,m)
B(1,0)	B(1,1)	B(1,2)	- - - - - -	B(1,m)
B(2,0)	B(2,1)	B(2,2)	- - - - - -	B(2,m)
- - -	- - -	- - -	- - - - - - -	- - -
B(n,0)	B(n,1)	B(n,2)	- - - - - -	B(n,m)

Questa matrice, a due dimensioni, ha n+1 righe e m+1 colonne. Matrici a una dimensione sono utilizzate normalmente per tenere il contenuto di una lista, mentre matrici a due dimensioni vengono tipicamente utilizzate per le tabelle.

Definizione delle matrici

Una matrice può essere definita usando l'istruzione dichiarativa DIM e specificando gli indici massimi da usare. Se la matrice deve contenere variabili stringa, ciò deve essere specificato con il simbolo $ al momento della definizione; ad esempio:

```
10 DIM A$(5,5), B(12) <ENTER>
```

specifica una matrice a due dimensioni di variabili stringa in cui gli indici vanno da 0,0 a 5,5 , ed una matrice a una dimensione con indici da 0 a 12.

Se in un programma viene menzionato un elemento di matrice, e questa non è stata definita in precedenza, ne viene creata una con il numero di dimensioni richieste dall'elemento e con un indice massimo per ogni dimensione pari a 10.

L'istruzione DIM inizializza anche tutti gli elementi della matrice a zero o a stringa nulla. Non vi sono comandi separati per l'azzeramento di matrici.

ARITMETICA BASIC

Le operazioni in ambiente BASIC si suddividono in tre categorie: relazionali (o di confronto), numeriche e logiche. Ogni categoria raggruppa un certo numero di funzioni, ciascuna identificata da un proprio simbolo chiamato operatore.

OPERATORI DI CONFRONTO (RELAZIONALI)

L'operazione più semplice è il confronto fra due espressioni. Le espressioni devono essere dello stesso tipo; non è possibile cioè il confronto tra stringhe e numeri. Gli operatori di confronto sono utilizzabili sia per dati numerici che per stringhe di caratteri. Le stringhe devono essere della stessa lunghezza, altrimenti la stringa più lunga viene troncata per poter essere raffrontata alla più corta. Il 'valore' di una stringa è determinato dal codice ISO di ciascun

carattere, il primo essendo il più significativo.

La tabella riportata in Figura 12-2 elenca gli operatori di confronto ed il loro significato.

OPERATORE	SIGNIFICATO
=	Uguale a
>	Maggiore di
<	Minore di
> = o = <	Maggiore o uguale a
< = o = >	Minore o uguale a
< > o > <	Non uguale a

Fig. 12-2 Operatori di confronto (relazionali)

L'operazione di confronto fornisce un risultato numerico. Esso può essere -1 (se il confronto è vero) oppure 0 (se invece è falso).

OPERATORI NUMERICI

Operazioni numeriche sono possibili solo su numeri. Il simbolo di addizione (+) può essere usato anche con le stringhe, per concatenarle, ma non si possono usare gli altri operatori numerici. La tabella riportata in Figura 12-3 elenca gli operatori disponibili e il loro significato.

OPERATORI	SIGNIFICATO
+	Addizione
-	Sottrazione o cambio di segno
*	Moltiplicazione
/	Divisione
ç	Divisione tra interi
^	Elevamento a potenza

Fig. 12-3 Operatori numerici

I numeri usati con un operatore sono detti operandi.

Nella divisione tra interi, gli operandi sono arrotondati al numero intero più vicino prima di effettuare la divisione e anche il quoziente viene troncato al valore intero più vicino. Il resto di tale divisione può essere ottenuto, sempre come valore intero più vicino, usando la funzione MOD.

Il livello di priorità per gli operatori numerici è il seguente:

1. Elevamento a potenza

2. Cambio di segno

3. Moltiplicazione e divisione

4. Divisione tra interi

5. Addizione e sottrazione

Per gli operatori con lo stesso livello di priorità, le operazioni vengono eseguite nell'ordine da sinistra a destra. Le priorità possono essere modificate usando le parentesi, seguendo le regole dell'algebra, per racchiudere le operazioni che devono essere svolte per prime. Ciò può essere esteso, usando parentesi entro altre parentesi, come illustrato negli esempi seguenti:

1.	Per	$\frac{x + y + z}{2}$	si scrive	(X + Y + Z)/2
2.	Per	$2x + 5$	si scrive	2 * X + 5
3.	Per	$2(x + 5)$	si scrive	2 * (X + 5)
4.	Per	$(x + y)^2$	si scrive	(X + Y)^2
5.	Per	$\frac{x^2(x + y)^4}{z^4}$	si scrive	X^2 * ((X + Y)/Z)^4

Le parentesi possono essere usate anche dove non sono strettamente necessarie. Ciò può essere utile per una maggior chiarezza.

Se un'espressione contiene una variabile cui non è stato precedentemente assegnato alcun valore, ad essa verrà automaticamente attribuito il valore zero.

Risultati dei calcoli

Il tipo di un'espressione numerica, e cioè il tipo del risultato ottenuto dallo svolgimento di un'espressione, dipende dal tipo degli operandi. Un'espressione in cui intervengano più operandi può essere considerata come una serie di calcoli in cui intervengono due operandi. Ad esempio nell'espressione:

2 * (X + 5) per primo viene calcolato (X + 5), ottenendo Y; quindi viene calcolato (2 * Y).

Il risultato di ciascuna operazione elementare, se i due operandi hanno precisione diversa, assume la forma di maggior precisione; cioè se un operando è in doppia precisione e l'altro è un intero, il risultato sarà in doppia precisione.

Gli operandi possono essere espressi in notazione scientifica, ad es.:

5.26E4 rappresenta 5.26×10^4

Se il valore di un risultato è maggiore del massimo permesso per quel tipo di dato verrà visualizzato un messaggio di overflow (OV). Un tentativo di divisione per zero porterà allo stesso risultato, con il messaggio d'errore /0. Questo messaggio appare anche quando zero è elevato a potenza negativa. Se zero è elevato a zero il risultato è 1.

Se il valore del risultato è minore del minimo consentito, viene visualizzato come zero.

OPERATORI LOGICI

In BASIC si possono effettuare operazioni logiche, utilizzando gli operatori booleani riportati - in ordine di priorità - nell'elenco che segue:

- NOT
- AND
- OR
- OR esclusivo (XOR)
- Implicazione (IMP)
- Equivalenza (EQV)

La tabella di verità per queste funzioni è riportata in Figura 12-4.

	A 0 B 0	A 0 B 1	A 1 B 0	A 1 B 1
NOT A	1	1	0	0
A AND B	0	0	0	1
A OR B	0	1	1	1
A XOR B	0	1	1	0
A EQV B	1	0	0	1
A IMP B	1	1	0	1

Fig. 12-4 Operatori logici

Gli operatori logici possono essere usati in congiunzione con quelli di confronto. In un'espressione le operazioni logiche sono svolte, dopo quelle numeriche e quelle di confronto, sui valori binari degli operandi. Se i valori binari sono esterni all'intervallo da -32768 a +32767, viene segnalato errore. L'operazione viene eseguita bit per bit ed il risultato è fornito come valore decimale entro lo stesso intervallo. I numeri binari negativi sono espressi nella forma "complemento a 2". Gli esempi che seguono illustrano come si possono usare gli operatori logici. L'uso di due operatori logici consecutivi è consentito solo se il secondo è NOT.

Esempio 1

In BASIC, impostare il comando:

```
? 4 OR 2 <ENTER>
 6
Ok
```

La prima riga è il comando, la seconda il risultato visualizzato da M10 e la terza il messaggio BASIC.

```
In binario, 4 = 100
            2 = 010
e           6 = 110
```

che è il risultato dell'operazione di OR su ogni coppia di bit.

Esempio 2

Impostare il comando:

```
? 7 AND 10 <ENTER>
 2
Ok
```

La prima riga è ancora il comando, la seconda il risultato visualizzato dal sistema e la terza il messaggio BASIC.

```
In binario,  7 = 0111
            10 = 1010
e            2 = 0010
```

che è il risultato dell'operazione di AND su ogni coppia di bit.

Le espressioni logiche possono essere utilizzate per verificare una condizione nelle istruzioni IF.. GOTO.. ELSE oppure IF.. THEN.. ELSE.

INTRODUZIONE DEI DATI

I dati possono essere introdotti come parte del programma, usando le istruzioni LET, DATA, READ e RESTORE. E' altresì possibile introdurre dati durante l'esecuzione del programma. Ciò si ottiene utilizzando le istruzioni INPUT o LINE INPUT sia per i dati introdotti da tastiera, che per i dati introdotti tramite file; il formato è diverso nei due casi (vedere al Capitolo 15).

13. CARATTERISTICHE DI PROGRAMMAZIONE

SALTI

Nei programmi BASIC, l'istruzione GOTO n provoca un trasferimento di controllo (salto) incondizionato alla riga n, modificando così la normale sequenza di esecuzione.

Per trasferimenti di controllo (salti) condizionati sono disponibili le seguenti istruzioni:

IF.. THEN.. ELSE..
IF.. GOTO.. ELSE..
ON.. GOTO..

Queste istruzioni possono essere 'nidificate' (viene così definito il processo di introduzione di una attività all'interno di un'altra attività); il formato di queste istruzioni è descritto nel Capitolo 15.

CICLI ITERATIVI (LOOP)

L'esecuzione ripetuta di una serie di istruzioni (loop di programma) può essere ottenuta usando l'istruzione FOR.. NEXT . Anche i loop possono essere nidificati come i salti, con il vincolo della capacità di memoria. Se i cicli nidificati hanno lo stesso punto finale, si può utilizzare per tutti una sola istruzione NEXT. Non vi è segnalazione di errore se un'istruzione FOR non ha il corrispondente NEXT, ma se un'istruzione NEXT viene eseguita prima della corrispondente istruzione FOR viene emesso il messaggio d'errore NF.

SOTTOPROGRAMMI

Quando una stessa sequenza di istruzioni deve essere eseguita più di una volta nell'ambito di uno stesso programma, si può scrivere un sottoprogramma che può essere richiamato in qualsiasi punto del programma stesso. L'istruzione GOSUB n provoca un salto di programma alla riga n , e la continuazione di esso con l'esecuzione di un sottoprogramma BASIC (BASIC subroutine) che inizia alla riga n. L'ultima istruzione della subroutine deve essere un'istruzione RETURN, che ritrasferisce il controllo al programma principale (alla prima riga successiva a GOSUB n).

SOTTOPROGRAMMI IN LINGUAGGIO MACCHINA

In un programma BASIC si può richiamare un sottoprogramma in linguaggio macchina per mezzo dell'istruzione CALL, che chiama il particolare programma desiderato. Per le limitazioni, vedere CALL al Capitolo 15.

14. FILE

I file possono contenere programmi o dati. Ai file contenenti programmi BASIC viene automaticamente assegnato un suffisso .BA, mentre i file testo e i file contenenti dati hanno un suffisso .DO. I file di programmi in linguaggio macchina hanno un suffisso .CO. I file dati hanno un'organizzazione sequenziale. Generalmente essi vengono creati in memoria; quindi possono essere trasferiti su cassetta o altra periferica. E' possibile scrivere un file direttamente su cassetta, ma non accodare (aggiungere) dati ad un file su cassetta. Quando si registra un file su cassetta, si ricopre sempre l'eventuale originale precedente. Per leggere o scrivere un file dati occorre prima aprirlo, usando l'istruzione OPEN.

APERTURA DI UN FILE

L'istruzione OPEN, oltre ad aprire il file, specifica anche la modalità del trasferimento dei dati. Per leggere un file è necessario aprirlo in input, per scrivere su un file è necessario aprirlo in output.

I file dati in memoria possono essere aperti anche in modo 'append': tutti i dati scritti sul file sono aggiunti ai precedenti, conservando pertanto anche i dati originali. Un file in memoria può essere registrato su cassetta usando l'istruzione SAVE, mentre un file su cassetta può essere caricato in memoria per mezzo dell'istruzione LOAD.

Con l'istruzione OPEN, viene assegnato al programma o al file dati un numero di file. Tale numero rimane associato al file, con la modalità specificata, per tutto il tempo in cui il file resta aperto; esso non può essere utilizzato per un'altra modalità se prima il file non è stato chiuso.

LETTURA DI UN FILE DATI

I dati di un file possono essere usati in un programma, aprendo prima il file in input, e utilizzando poi l'istruzione INPUT per estrarre i dati desiderati e assegnarli a variabili di programma.

Per individuare la fine del file occorre usare l'istruzione EOF dopo la INPUT. Se ciò non viene fatto, una volta raggiunta la fine del file sarà visualizzato un messaggio d'errore EF e il programma terminerà in modo anomalo.

SCRITTURA DI UN FILE DATI

I dati possono essere scritti su un file previa apertura dello stesso in modalità output, usando le istruzioni PRINT o PRINT USING.

CHIUSURA DI UN FILE

Alcune istruzioni e comandi provocano la chiusura di tutti i file aperti, ad esempio END. Se si vogliono chiudere file indipendentemente da tali istruzioni, occorre usare l'istruzione CLOSE.

15. ELENCO DEI COMANDI, FUNZIONI E ISTRUZIONI BASIC

ABS Funzione

ABS calcola il valore assoluto (positivo) di un'espressione numerica.

Formato: ABS(x)

dove: x è un'espressione.

Esempio:

```
Ok
PRINT ABS(6*(-3))
 18
Ok
```

ASC Funzione

ASC restituisce il codice ISO (in decimale) del primo carattere di un'espressione stringa.

Formato: ASC(x$)

dove: x$ è un'espressione stringa.

Se x$ è la stringa nulla, viene visualizzato un codice d'errore FC (chiamata irregolare di una funzione). Vedere anche la funzione CHR$ per convertire un codice ISO in una stringa.

Esempio:

```
Ok
10 X$ = "M10"
20 PRINT ASC(X$)
RUN
 77
Ok
```

ATN Funzione

ATN calcola l'arcotangente di un numero.

Formato: ATN(x)

dove: x è un numero o un'espressione numerica qualsiasi.

Il risultato è dato in radianti, nell'intervallo $-\pi/2$, $+\pi/2$.

BEEP Istruzione

BEEP provoca una segnalazione acustica della durata di circa mezzo secondo.

Formato: BEEP

Esempio: IF X = 0 THEN BEEP

Questa istruzione può anche essere usata in un ciclo FOR/NEXT per allungare il suono, o in un ciclo GOTO per segnalare la fine di un lungo programma.

Esempio:
```
 990  BEEP
1000  GOTO 990
```

Vedere anche l'istruzione SOUND.

CALL Istruzione

CALL richiama una subroutine in linguaggio macchina.

Formato: CALL indirizzo, A ,HL

dove: 'indirizzo' è l'indirizzo di partenza in ROM della subroutine in linguaggio macchina

A è un intero da introdurre nel registro A

HL è un intero da introdurre nel registro HL.

L'indirizzo di partenza deve essere compreso fra -23758 e 65535. Gli interi da assegnare ai registri devono essere compresi fra 0 e 255.

CDBL Funzione

CDBL converte un numero in doppia precisione.

Formato: CDBL(x)

dove: x è un intero o un numero in semplice precisione.

Nel Capitolo 12 viene chiarito come sono eseguite le conversioni. Vedere anche CINT e CSNG per conversioni a numero intero o in semplice precisione.

CHR$ Funzione

CHR$ restituisce un carattere ISO a fronte di un intero tra 0 e 255.

Formato: CHR$(i)

dove: i è un intero o un'espressione intera il cui valore è compreso tra 0 e 255.

Esempio:

```
Ok
PRINT CHR$(90)
Z
Ok
```

CINT Funzione

CINT converte un numero in un intero eliminando tutte le cifre dopo la virgola decimale.

Formato: CINT(x)

dove: x è un numero o un'epressione in semplice o doppia precisione.

Il numero viene troncato con la perdita delle cifre decimali. Se il risultato non è compreso nell'intervallo -32768 e 32767 viene evidenziato un errore OV (overflow). Le conversioni sono spiegate nel Capitolo 12. Vedere anche CDBL e CSNG per la conversione in doppia e singola precisione.

Esempio:

```
Ok
PRINT CINT(96.34268)
 96
Ok
```

CLEAR Comando

CLEAR pulisce lo spazio di memoria usato per i dati.

Formato: CLEAR 'spazio per stringhe', 'locazione più alta'

dove: 'spazio per stringhe' è il numero di byte da rendere disponibile per variabili stringa

'locazione più alta' è la più alta locazione di memoria disponibile per il BASIC. Se si specifica MAXRAM l'intera memoria è disponibile.

Il valore di default dello spazio per stringhe è 256 byte. CLEAR inizializza tutte le variabili numeriche a 0 e tutte le stringhe a stringa nulla. Vengono anche chiusi tutti i file aperti.

Si noti che lo spazio per le stringhe viene usato dal sistema solo se non vi è altro spazio disponibile per le stringhe, cioè in un programma lo spazio per le variabili sarà allocato automaticamente e non vi sarà necessità di richiedere spazio in memoria mediante comando CLEAR.

CLOAD , CLOAD? , CLOADM Comandi

CLOAD e CLOADM vengono descritti con il comando LOAD. CLOAD? viene usato per confrontare un programma BASIC su cassetta con uno in memoria.

Formato: CLOAD? "nomefile"

dove: nomefile è il nome con cui il file è stato salvato.

Questo comando verifica che il programma sia stato salvato correttamente. Esso può essere usato solo per programmi salvati in formato binario.

Il programma su nastro viene confrontato con quello in RAM, byte per byte. Se viene trovata qualche differenza, il trasferimento dati su nastro non è accettabile e viene emesso il messaggio **Verify failed.** In questo caso il programma deve essere salvato nuovamente.

CLOSE Istruzione

CLOSE conclude l'accesso a una periferica o a un file.

Formato: CLOSE file-numero-1 ,file-numero-2 ...

dove: il numero del file è quello con il quale esso è stato aperto (vedere istruzione OPEN). Con una sola istruzione si può chiudere un numero qualunque di file.

Tutte le periferiche del sistema saranno chiuse una volta eseguita l'istruzione CLOSE. Tutti i file saranno chiusi automaticamente nelle seguenti condizioni:

- se sono eseguiti un'istruzione END e un comando NEW
- se sono eseguiti un'istruzione RUN e un comando LOAD
- se il programma viene cambiato
- se viene eseguita una CLOSE senza operandi.

CLS Comando

CLS serve a cancellare il display.

Formato: CLS

Viene ripulito l'intero display a meno che sull'ultima riga non siano presenti le etichette dei tasti funzione. In questo caso vengono ripulite solo le prime sette righe.

COM ON/OFF/STOP Istruzioni

COM ON/OFF/STOP servono ad abilitare o disabilitare il colloquio tramite interfaccia RS-232C.

Formato:
```
COM ON
COM OFF
COM STOP
```

COM ON abilita la possibilità di sondare l'interfaccia di comunicazione. Se si utilizza l'istruzione ON COM GOSUB per cedere il controllo a una subroutine, il BASIC verifica il buffer di comunicazione a ogni linea di programma. Se è arrivato qualche nuovo carattere, il controllo passa alla subroutine.

COM OFF disabilita la subroutine.

COM STOP inibisce la chiamata alla routine fino a che non viene eseguito un successivo comando COM ON. Se arriva qualche carattere mentre è attivo un COM STOP, la subroutine viene chiamata appena viene eseguita un successivo COM ON.

Vedere anche ON ... GOSUB per le relative informazioni.

CONT Comando

CONT è usato per proseguire l'esecuzione di un programma dopo un'interruzione dovuta alla pressione di <BREAK> o all'istruzione STOP.

Formato: CONT

Questo comando è usato soprattutto in 'debugging'. Se si modifica un programma durante un'arresto di esecuzione, un eventuale tentativo di farlo proseguire provoca la segnalazione di errore CN (cannot continue = non posso continuare).

COS Funzione

COS calcola il coseno di un angolo.

Formato: COS (x)

dove: x è un angolo espresso in radianti.

Se x è noto in gradi, si deve moltiplicarlo per la costante 0.01745329 per convertirlo in radianti.

Esempio:

```
PRINT COS(60*.01745329)
 .50000013094004
Ok
```

CSAVE , CSAVEM Comandi

Vedere il comando SAVE.

CSNG Funzione

CSNG converte un numero a semplice precisione.

Formato: CSNG(x)

dove: x è un intero o un numero in doppia precisione.

Nel Capitolo 12 è spiegato come vengono eseguite le conversioni. Vedere anche CINT e CDBL per conversioni a intero e a doppia precisione.

CSRLIN Funzione

CSRLIN restituisce la posizione verticale del cursore.

Formato: CSRLIN

Questa funzione può essere usata per il display LCD, nel qual caso può assumere un valore da 0 a 7, o per un eventuale video collegato, nel qual caso potrà assumere un valore da 0 a 24. Il valore 0 rappresenta la prima riga del display.

DATA Istruzione

DATA memorizza le costanti numeriche e stringa cui si può accedere con l'istruzione READ.

Formato: DATA costante-1,costante-2 ...

dove: costante-n è una costante numerica o stringa.

Si possono specificare più costanti, fino al completamento della linea programma. Le costanti stringa non devono necessariamente essere racchiuse tra virgolette, a meno che non contengano virgole, punti o spazi significativi iniziali o finali. Vedere anche l'istruzione READ.

Esempio:

```
Ok
10 READ A,B$,C,D$
20 PRINT A,B$,C,D$
30 DATA 56,AB,-3.55E+20,6abc7
RUN 10
 56                AB
-3.55E+20          6abc7
Ok
```

DATE$ Istruzione e Funzione

DATE$ specifica la data sull'orologio o restituisce la data attuale.

Formato:

```
DATE$ = "gg/mm/aa"
DATE$
```

dove:

gg è il giorno, compreso tra 01 e 31.

mm è il mese, compreso tra 01 e 12.

aa è l'anno, compreso tra 01 e 99.

Se DATE$ viene usato come funzione (es. PRINT DATE$) il risultato è la stampa della data attuale. Alla prima accensione il sistema stabilisce sul menù la data 01/01/1900.

DAY$ Istruzione e Funzione

DAY$ specifica il giorno della settimana sull'orologio, o restituisce il giorno attuale.

Formato:

```
DAY$ = "xxx"
DAY$
```

dove:

xxx è una tra le seguenti stringhe: Mon (lunedì), Tue (martedì), Wed (mercoledì), Thu (giovedì), Fri (venerdì), Sat (sabato), Sun (domenica).

Se DAY$ viene usato come funzione (es. PRINT DAY$) il risultato è la stampa del giorno attuale. Alla prima accensione il sistema è posizionato sul Sun (domenica).

DIM Istruzione

DIM specifica, in una matrice, il numero delle dimensioni e il numero di elementi per ciascuna dimensione.

Formato: DIM x(dim1 ,dim2 ...,dimn), y(dim1 ,dim2 ...,dimn) ,z(dim1 ,dim2 ...,dimn)

dove: x,y,z sono variabili stringa o numeriche.

dim1 dim2 ecc. sono interi che specificano il numero di elementi per ciascuna dimensione.

Si noti che il computo degli elementi parte da zero, per cui la matrice A(1) è a una dimensione e ha due elementi, etc. Le matrici possono avere qualunque numero di dimensione e ogni dimensione può avere qualunque numero di elementi, purchè vi sia spazio sufficiente in memoria. Se ci si riferisce a una matrice senza specificare un'istruzione DIM, il numero di elementi per ogni dimensione viene assunto uguale a 10.

L'istruzione DIM inizializza tutti gli elementi della matrice specificata al valore iniziale 0 o stringa nulla.

Esempio:

```
10 DIM A(5),B(2,3),C$(6,7)
```

Questa istruzione specifica una matrice a una dimensione e 6 elementi, una a due dimensioni rispettivamente di 3 e 4 elementi ed una matrice stringa a due dimensioni di 7 e 8 elementi.

EDIT Comando

EDIT cambia la modalità operativa da BASIC a TEXT per eseguire operazioni di editing sul programma attuale. Si ritorna in BASIC premendo <F8>.

Formato: EDIT n -m

dove: n è il numero della prima riga da modificare

m è il numero dell'ultima riga da modificare.

Se non si specificano i numeri di riga, l'intero programma è reso disponibile. Tutte le caratteristiche di TEXT sono utilizzabili in EDIT, tranne la prestazione di riposizionamento a capo automatico che non è attiva. Il programma TEXT è descritto nel Capitolo 5.

END Istruzione

END termina l'esecuzione di un programma, chiude tutti i file aperti e riporta il sistema in Stato Comandi.

Formato: END

Questa istruzione non provoca l'emissione di alcun messaggio; essa può essere posta ovunque nel programma, per terminarne l'esecuzione in un punto diverso dall'ultima istruzione. L'istruzione END come ultima istruzione di un programma è opzionale.

Esempio:

```
10 INPUT A,B
20 GOSUB 100
.
.
90 END
100 C = SQR (A*A + B*B)
110 IF A<B GOTO 120 ELSE GOTO 140
120 D = SQR (A*A - B*B)
130 RETURN
140 D = SQR (B*B - A*A)
150 RETURN
```

EOF Funzione

EOF sonda la fine di un file.

Formato: EOF(n)

dove: n è il numero del file specificato nella OPEN

Questa funzione è usata dopo un'istruzione OPEN e prima dell'istruzione INPUT/OUTPUT per un file determinato. Restituisce -1 se è stata raggiunta la fine del file, altrimenti restituisce 0. Viene utilizzata per eseguire un test di fine file ed evitare che il sistema evidenzi un errore EF (end of file error = errore di fine file). Può essere usata per qualsiasi file sequenziale o per i dati trasferiti via modem o interfaccia RS-232C.

Esempio:

```
10 OPEN "RAM:DATA" FOR INPUT AS £1
20 IF EOF(1) GOTO 100
30 INPUT £1, A$,B
.
.
90 GOTO 20
100 CLOSE £1
```

ERL/ERR Funzioni

ERL e ERR forniscono il numero di riga ed il numero di identificazione di un errore, per utilizzarli in una routine di gestione dell'errore stesso.

Formato:
```
IF ERL = num-riga THEN ...
IF ERR = num-err  THEN ...
```

dove: num-riga è il numero della riga che ha provocato l'ultimo errore.

num-err è il numero dell'ultimo errore (vedere in Appendice B la lista completa degli errori con relativi codici e numeri).

Queste funzioni vengono normalmente usate in istruzioni IF ... THEN per dirottare il controllo del programma ad una routine di gestione degli errori. Se l'errore è in una riga immediata, ERL conterrà il numero 65535: per eseguire il test si dovrà scrivere:

```
IF 65535 = ERL THEN ...
```

Per altre informazioni sul trattamento degli errori vedere ERROR.

ERROR Istruzione

ERROR è usata per simulare una condizione d'errore o definire un errore.

Formato: ERROR n

dove: n è un intero tra 0 e 255.

Se n è il numero di un errore esistente viene emesso un messaggio.

Esempio:
```
ERROR 5
?FC ERROR
Ok
```

Le routine di trattamento degli errori vengono richiamate dall'istruzione ON ERROR GOTO n , dove n è la prima riga di tale routine. Per eliminare la gestione dell'errore, usare l'istruzione ON ERROR GOTO 0.

Per definire un codice d'errore, scegliere un numero che non sia già assegnato. E' consigliabile usare numeri che siano alla fine dell'intervallo, per evitare sovrapposizioni con i codici assegnati al BASIC in caso di future espansioni. Se non vi sono routine di gestione a fronte dell'istruzione ERROR, o se non è stato definito nessun messaggio, viene visualizzato il messaggio UE (unprintable error = errore non stampabile).

Esempio:

```
40 ON ERROR GOTO 1000
.
.
90  INPUT  A
100 IF  A < 0 THEN ERROR 250
.
.
1000 IF ERR = 250 THEN PRINT "Deve essere positivo"
```

EXP Funzione

EXP calcola una funzione esponenziale.

Formato: EXP(x)

dove: x è un numero nell'intervallo fra -149.664 e 145.0628. Se questa condizione non è verificata viene emesso il messaggio OV (overflow = superamento).

Esempio:

```
Ok
PRINT EXP(0)
 1
Ok
```

FILES Comando

FILES visualizza i programmi e i file dati attualmente in RAM.

Formato: FILES

Ogni nome di file ha un suffisso che lo classifica come segue:

- .BA è un file programma BASIC
- .CO è un file programma in linguaggio macchina
- .DO è un file dati creato da TEXT o un programma BASIC memorizzato in formato ISO

I file presenti in RAM sono visualizzati anche premendo il tasto <F1>.

FIX Funzione

FIX calcola la parte intera di un numero, troncando la parte dopo il separatore decimale.

Formato: FIX (x)

dove: x è un numero nella gamma permessa.

Esempio:

```
Ok
FIX(3.9)
 3
Ok
FIX(-3.9)
-3
Ok
```

Si noti che INT(-3.9) o INT(-3.2) darebbero come risultato -4.

FOR...NEXT Istruzioni

Le istruzioni FOR e NEXT consentono di eseguire una serie di istruzioni, ciclicamente, per un determinato numero di volte.

Formato:

```
FOR a = n TO p  STEP s
.
.
NEXT  z,y ..., a
```

dove: a,y,z, sono variabili

n,p,s sono espressioni numeriche

La variabile a è un contatore il cui valore iniziale è n ed è incrementato di s ad ogni ciclo. Le linee di programma contenute tra FOR e NEXT vengono eseguite ciclicamente sino a che a = p + s. Il programma prosegue con la linea successiva all'istruzione NEXT. Ciò significa che il ciclo è eseguito (p-n)/s volte. Se s non è specificato, viene assunto =1. Si possono usare passi negativi, purchè il valore di n sia maggiore del valore di p. Nel caso si scelga un insieme di valori incoerenti, ad esempio se s è maggiore della differenza tra n e p, la sequenza viene eseguita una sola volta. Se si assegna zero ad s, il ciclo prosegue ininterrottamente.

Le sequenze FOR...NEXT possono essere nidificate l'una nell'altra. Nel formato specificato sopra, y e z sono contatori ciclici per cicli

secondari, nidificati all'interno del ciclo principale. La sola limitazione riguardante il numero di sequenze nidificate è data dalla quantità di memoria disponibile. Ciascuna sequenza deve essere completata con una NEXT e la sequenza più interna deve essere terminata per prima. Se i cicli terminano nello stesso punto, è possibile combinare le diverse variabili in una sola istruzione NEXT, scrivendo per prima la variabile del ciclo più interno. Se un'istruzione NEXT non si riferisce ad alcuna variabile, viene assunta la variabile dell'ultima istruzione FOR. Se si incontra un'istruzione NEXT prima di una FOR, viene emesso l'errore NF (Next senza For). Perciò un salto dentro un ciclo FOR NEXT causerà un errore mentre un salto fuori dal ciclo non ne causerà.

Esempio:

```
10  INPUT  J
20  FOR  I = 1 TO J
30  PRINT I;
40  FOR  K = 1 TO 3
50  PRINT "a";
60  NEXT K ,I
80  GOTO 10
RUN
? 4
 1 aaa 2 aaa 3 aaa 4 aaa?
```

FRE Funzione

FRE calcola il numero di byte di memoria disponibile per l'utente.

Formato:

```
FRE(x)
FRE(x$)
```

dove:

x è un argomento numerico fittizio; normalmente 0.

x$ è un argomento stringa fittizio; normalmente la stringa nulla, "".

Se FRE viene usata con argomento numerico BASIC, restituisce il numero di byte in memoria disponibili per programmi BASIC, file testo, programmi in linguaggio macchina, ecc. Se viene invece usata con argomento stringa, restituisce la quantità di memoria disponibile per le stringhe. Lo spazio per le stringhe è liberato dalla precedente istruzione CLEAR.

Esempio:

```
Ok
PRINT FRE("")
 200
Ok
```

In questo esempio lo spazio per le stringhe è di 200 byte.

GOSUB e RETURN Istruzioni

GOSUB e RETURN sono usate per trasferire il controllo del programma a una subroutine, al termine della quale viene ritrasferito al programma principale.

Formato: GOSUB n

RETURN

dove: n è il numero della prima riga della subroutine.

GOSUB provoca un salto del programma alla subroutine specificata e RETURN trasferisce il controllo del programma alla prima istruzione successiva alla GOSUB. Una subroutine può contenere, se necessario, più di un'istruzione RETURN.

Una subroutine sarà preceduta da un'istruzione STOP, END o GOTO, per evitarvi un accesso non programmato, se non con un'istruzione GOSUB. Se viene incontrato un RETURN senza un precedente GOSUB verrà evidenziato un errore RG (RETURN senza GOSUB). Salti all'interno di una subroutine causeranno errore, mentre salti fuori da essa non lo provocheranno.

Le subroutine possono essere nidificate una nell'altra. Questo procedimento è limitato soltanto dalla memoria disponibile.

Esempio:

```
100  GOSUB 200
110  PRINT "Rientro da routine esterna"
.
190  END
200  PRINT "Esecuzione routine esterna"
.
220  GOSUB 300
230  PRINT "Rientro da routine interna"
.
250  RETURN
.
300  PRINT "Esecuzione routine interna"
.
320  RETURN
RUN

Esecuzione routine esterna
Esecuzione routine interna
Rientro da routine interna
Rientro da routine esterna

Ok
```

Vedere anche ON...GOSUB

GOTO Istruzione

GOTO viene usata per eseguire un salto incondizionato ad una riga di programma, al di fuori della normale sequenza.

Formato: GOTO n

dove: n è un numero di riga

Può essere usata in una riga immediata per far partire un programma da un punto prefissato. Normalmente viene usata all'interno di un programma per trasferire il controllo a una data riga.

Esempio:

```
350  GOTO 10
```

Verrà effettuato un salto incondizionato dalla riga 350 alla 10. Per i salti condizionati, vedere le istruzioni IF...THEN...ELSE e ON...GOTO.

HIMEM Funzione

HIMEM ritorna l'indirizzo più alto di memoria disponibile per l'utente.

Formato: HIMEM

Il valore della funzione è determinato dall'ultima istruzione CLEAR. Se non ne sono state eseguite, il valore è 62960 (MAXRAM).

Esempio:

```
Ok
PRINT HIMEM
 40960
Ok
```

IF...GOTO/THEN...ELSE... Istruzioni

IF...GOTO...ELSE e IF...THEN...ELSE sono istruzioni usate per eseguire dei salti condizionati in un programma.

Formato:

```
IF x GOTO n ELSE yyy
IF X THEN yyy ELSE zzz
```

dove: x è un'espressione relazionale

yyy e zzz sono clausole che possono contenere istruzioni e numeri di riga.

Se l'espressione x è vera, il controllo del programma passa all'istruzione GOTO o alla clausola THEN; se è falsa, il controllo del programma passa alla clausola ELSE.

Esempio:

```
.
110  IF A > B GOTO 120 ELSE GOTO 130
120  D = SQR(A*A - B*B)
130  D = SQR(B*B - A*A)
.
```

Le istruzioni IF possono essere nidificate. La limitazione è la lunghezza della riga in quanto IF...GOTO/THEN...ELSE è un'unica istruzione e deve quindi completarsi entro una riga di programma.

Esempio:

```
50  IF A > 0 THEN PRINT "POSITIVO" ELSE IF A = 0
THEN PRINT "UGUALE" ELSE IF A < 0 THEN PRINT "NEGATIVO"
```

Se in un istruzione non vi è lo stesso numero di elementi THEN e ELSE, ogni ELSE viene associato al più vicino THEN. Ciò può portare a un'istruzione incompleta.

Esempio:

```
70 IF A = B THEN IF B = C THEN PRINT "A=C" ELSE PRINT "?"
```

non visualizzerà ? quando A ≠ B ma quando A = B e B ≠ C.

INKEY$ Funzione

INKEY$ restituisce una stringa di un carattere equivalente al tasto premuto sulla tastiera; nel caso non siano stati premuti tasti, viene restituita una stringa nulla.

Formato: INKEY$

Ogni tasto premuto restituisce un carattere, eccetto <CTRL> + C, che termina l'esecuzione del programma. Questa funzione viene spesso usata per fermare un programma fino alla digitazione di un tasto.

Esempio:

```
110 PRINT "Premere un tasto qualunque per proseguire"
120 A$ = INKEY$
130 IF A$ = "" GOTO 120 ELSE GOTO 140
140 ...
```

INP Funzione

INP acquisisce il byte presente ad una porta specificata.

Formato: INP(i)

dove: i è un intero tra 0 e 255 che identifica la porta.

INP è la funzione complementare di OUT.

INPUT Istruzione

INPUT usata per introdurre dati in un programma attraverso la tastiera.

Formato: INPUT "messaggio"; a , b, ... ,n

dove: messaggio è una frase guida che spiega il dato richiesto. Il sistema aggiunge ad essa un punto interrogativo; se il messaggio non è specificato, viene visualizzato solo quest'ultimo.

a,b,n sono variabili numeriche o stringa.

I dati introdotti devono corrispondere al tipo di variabile dichiarata, altrimenti viene visualizzato il messaggio **Redo from start** (rifare da capo). Se un'istruzione di INPUT richiede più di un dato, ogni dato introdotto deve essere separato dal successivo per mezzo di una virgola. Se i dati introdotti sono insufficienti, M10 visualizza **??**.

Esempio:

```
Ok
10 INPUT "Giorno e data del mese"; A$,B
20 PRINT "Oggi è ";A$;B
RUN
Giorno e data del mese? Sabato 14 (input dell'utente)
Oggi è Sabato 14
Ok
```

INPUT Istruzione

INPUT usata per introdurre dati in un programma da un file.

Formato: INPUT num-file,a ,b , ... ,n

dove: num-file è il numero col quale il file è stato aperto, preceduto dal simbolo £. Vedere OPEN

a,b,n sono variabili numeriche o stringa.

Gli elementi del file devono corrispondere alla lista delle variabili. I file possono essere in RAM, su cassetta, o provenienti da periferica connessa all'interfaccia di comunicazione.

Per dati numerici o stringhe, gli spazi iniziali, i comandi di ritorno a capo e di interlinea sono ignorati. Il primo carattere che non sia uno spazio, un ritorno a capo o un'interlinea viene considerato l'inizio del numero o della stringa. Un numero è completato quando si incontra uno spazio, un ritorno a capo, un'interlinea o una virgola. Per una stringa,

se essa inizia con ", viene considerata conclusa alle virgolette successive. Se invece il primo carattere non è ", tutti i caratteri ne fanno parte fino a quando non terminerà con un ritorno a capo, un'interlinea, una virgola o comunque dopo 255 caratteri.

INPUT$ Funzione

INPUT$ acquisisce una stringa di caratteri dalla tastiera o da un file.

Formato: INPUT$ (n, num-file)

dove: n è il numero di caratteri da leggere.

num-file è il numero con cui è stato aperto il file.

Il file può essere in RAM, su cassetta, o letto da periferica connessa all'interfaccia di comunicazione. Se non viene specificato alcun numero di file, i caratteri sono letti da tastiera.

INSTR Funzione

INSTR restituisce la posizione della prima ricorrenza di una stringa specificata, all'interno di un'altra stringa.

Formato: INSTR(i,x$,y$)

dove: i è un intero da 0 a 255 che rappresenta uno spiazzamento; se viene specificato, vengono tralasciati i caratteri prima di iniziare la ricerca della stringa richiesta.

x$ è la stringa da scandire

y$ è la sottostringa da cercare.

INSTR darà 0 in una delle seguenti condizioni:

- se i è maggiore della lunghezza di x$
- se y$ non viene trovata

- se x$ è nulla

- se x$ e y$ sono nulle.

Se solo y$ è nulla la funzione restituisce i oppure 1. Gli i caratteri iniziali esclusi dalla ricerca sono inclusi nel conteggio.

Esempio:

```
Ok
10  A$ = "Computer Portatile Olivetti M10"
20  B$ = "M10"
30  PRINT INSTR(9,A$,B$)
RUN
 29
Ok
```

INT Funzione

INT calcola il più grande intero minore o uguale a un numero dato.

Formato: INT(x)

dove: x è un numero di qualunque tipo

Esempio:

```
Ok
INT(3.9)
 3 Ok
INT(-3.9)
-4
Ok
```

Si noti che FIX(-3.9) darebbe -3.

IPL Comando

IPL è utilizzato per eseguire un programma immediatamente dopo l'accensione del sistema (solo partenza "a caldo").

Formato: IPL "nome-file"

dove: nome-file è il nome completo di un file.

Per cancellare un precedente comando di IPL si usa lo stesso comando, senza specificare l'argomento.

Esempio:
```
IPL "BASIC"
IPL "nome-file.BA"
```

Con questo comando M10 partirà in BASIC immediatamente dopo la visualizzazione del menù.

KEY Istruzione

KEY è utilizzata per assegnare una stringa a un dato tasto funzione. KEY LIST elenca i contenuti dei tasti funzione, mentre KEY ON/OFF/STOP attiva o disattiva la possibilità di saltare a una subroutine.

Formato:
```
KEY n, "xxx"
KEY LIST
KEY (n) ON
        OFF
        STOP
```

dove: n è un numero di tasto funzione (tra 1 e 8).

xxx è una stringa di lunghezza non superiore a 15.

I tasti funzione F1-F8 sono inizialmente associati ai valori mostrati nell'esempio che segue. L'utente può assegnare loro significati diversi con l'istruzione KEY n, "xxx".

Esempio:
```
KEY LIST
Files          Load "
Save "         Run
List
               Menù
Ok
```

La lista è nella forma seguente:

F1	F2
F3	F4
F5	F6
F7	F8

KEY (n) ON attiva l'entrata in una subroutine, allorchè viene premuto il tasto n. La subroutine deve essere specificata in una istruzione ON KEY GOSUB (vedere ON...GOSUB).

KEY (n) OFF disabilita la subroutine.

KEY (n) STOP impedisce di entrare nella subroutine sino al KEY (n) ON successivo. Se viene premuto un tasto funzione durante la fase attiva di KEY (n) STOP, la subroutine viene richiamata immediatamente dopo l'esecuzione di KEY (n) ON.

KILL Comando

KILL è usato per cancellare un file dalla RAM.

Formato: KILL "xxxxx.yy"

dove: xxxxx è il nome del file

yy è il suffisso, e cioè:

- .BA per un file programma BASIC
- .CO per un file programma in linguaggio macchina
- .DO per un file testo

Deve essere specificato l'intero nome; il file attualmente in uso non può essere cancellato.

LCOPY Istruzione

LCOPY lista il testo attualmente presente su display sulla stampante.

Formato: LCOPY

La stessa funzione può essere ottenuta manualmente premendo <PRINT>.

LEFT$ Funzione

LEFT$ restituisce una sottostringa di caratteri, a partire dalla sinistra e per una lunghezza assegnata, di una stringa specificata.

Formato: LEFT$(x$,i)

dove: x$ è una stringa

i è un intero tra 0 e 255

Vengono restituiti i primi i caratteri della stringa x$. Se i è maggiore della lunghezza di x$, viene restituita l'intera stringa; se i è 0, viene restituita una stringa nulla.

LEN Funzione

LEN restituisce la lunghezza di una stringa.

Formato: LEN(x$)

dove: x$ è una stringa.

Viene restituito il numero dei caratteri che compongono la stringa, compresi gli spazi e i caratteri non stampabili.

LET Istruzione

LET assegna un valore a una variabile.

Formato: LET x = a

dove: x è una variabile

a è un'espressione dello stesso tipo di x.

Un'espressione stringa non può essere assegnata a una variabile numerica e viceversa. La parola LET è opzionale: lo stesso risultato si ottiene con x = a.

LINE Comando

LINE è usato per tracciare una linea o un rettangolo sul display.

Formato: LINE (x1,y1) - (x2,y2) , c
LINE (x1,y1) - (x2,y2) , c , B F

dove: x1,y1 e x2,y2 sono le coordinate dei punti di partenza e di arrivo della linea; x deve essere compreso tra 0 (lato sinistro del display) e 239 (lato destro); y deve essere compreso tra 0 (bordo superiore) a 63 (bordo inferiore).

c è il parametro colore; se c=1, la linea o il rettangolo sono tracciati; se c=0, sono cancellati.

B (Box = scatola) è usato per ottenere un rettangolo, con i lati paralleli ai bordi del display, e la cui diagonale è la linea definita.

F (Fill = riempire) riempie il rettangolo definito con B.

Se le coordinate non sono comprese negli intervalli di cui sopra, viene emesso l'errore FC (illegal Function Call = chiamata funzione illegale).

Se le coordinate del primo punto non sono specificate, si assumono uguali alle ultime riferite nella precedente istruzione LINE, PRESET, PSET. In assenza di una precedente istruzione di questo tipo viene assunto (0,0).

Si noti che c è obbligatorio nel caso si definisca un rettangolo; nel caso di una linea, se c è omesso, viene assunto uguale a 1.

LINE INPUT Istruzione

LINE INPUT è usata per introdurre una stringa (lunghezza massima 254 caratteri) in una variabile, senza bisogno di delimitatori (").

Formato: LINE INPUT "messaggio"; a$

dove: messaggio viene visualizzato prima dell'introduzione della stringa; in questo caso non viene aggiunto il punto interrogativo, a meno che non faccia parte del messaggio.

a$ è la variabile stringa.

Tutti i caratteri introdotti da tastiera sono assegnati alla variabile fino alla digitazione di <ENTER>, che termina la linea. Se viene

introdotta la sequenza interlinea e ritorno carrello (LF+CR), il comando LF fa parte della variabile stringa, mentre CR è ignorato e l'input dei caratteri continuerà.

Per interrompere una LINE INPUT, premere <CTRL> + C o <BREAK>. Il BASIC ritorna allora in Stato Comandi e visualizza **Ok**. Impostando il comando CONT, il programma riprende l'esecuzione dall'istruzione LINE INPUT.

LINE INPUT Istruzione

LINE INPUT è usata per introdurre una stringa (lunghezza massima 254 caratteri) in una variabile stringa, da un file sequenziale.

Formato: LINE INPUT num-file, a$

dove num-file è il numero con il quale il file è stato aperto con OPEN, preceduto dal simbolo £.

a$ è la variabile stringa.

Questa istruzione legge tutti i caratteri del file fino ad incontrare un CR. La sequenza CR+LF è ignorata e la successiva LINE INPUT legge i caratteri fino al successivo CR. Eventuali sequenze LF+CR presenti nel file vengono trasferite alla stringa.

Questa istruzione può venire usata con file in RAM o su cassetta, o anche con i dati sequenziali di un buffer ricevuto via interfaccia di comunicazione.

LIST Comando

LIST visualizza tutto o parte del programma attuale.

Formato: LIST n - m

dove: n è il numero della prima linea da visualizzare

m è il numero dell'ultima linea da visualizzare.

Se non si specificano numeri di linea, vien fatta la lista dell'intero programma. Se n è omesso, vengono visualizzate le linee dall'inizio fino ad m; se è omesso m, vengono visualizzate le linee da n fino alla fine.

Per terminare LIST, premere <CTRL> + C o <BREAK>.

Per interrompere temporaneamente, premere <CTRL> + S o <PAUSE>; ripremere per continuare.

LLIST Comando

LLIST esegue una lista di tutto o parte di un programma sulla stampante. Il formato del comando è lo stesso del comando LIST.

LOAD , LOADM , CLOAD , CLOADM Comandi

LOAD, LOADM, CLOAD e CLOADM sono usati per caricare programmi nell'area di memoria di esecuzione, ma senza registrarli nella directory. Per registrare un programma nella directory è necessario memorizzarlo col comando SAVE appropriato.

Formato:

```
CLOAD "nome-file" , R
LOAD "CAS: nome-file" , R
LOAD "RAM: nome-file" , R
LOAD "COM:cbpsx"  ,R
CLOADM "nome-file"
LOADM "CAS: nome-file"
LOADM "RAM: nomefile"
```

dove: CLOAD e CLOADM caricano programmi da cassetta.

nome-file è il nome usato alla registrazione.

R fa eseguire il programma dopo il caricamento.

CAS carica il file programma da cassetta.

RAM carica il file programma da RAM.

COM carica il programma da linea di comunicazione; per i parametri vedere sotto.

CLOAD e LOAD caricano programmi BASIC nell'area di memoria di esecuzione, cancellando il programma eventualmente presente. CLOADM e LOADM caricano programmi in linguaggio macchina, memorizzati nella locazione specificata all'atto della registrazione.

Normalmente un comando LOAD chiude tutti i file. L'opzione R lancia l'esecuzione del programma a caricamento avvenuto e tiene aperti i file dati. Usando LOAD con opzione R è possibile accodare un programma a un altro. Vedere anche il comando RUN.

I comandi CLOAD e LOAD "CAS: hanno la medesima funzione. Entrambi caricano il programma specificato da cassetta di nastro magnetico. Se viene omesso il nome del programma, viene caricato il primo programma BASIC trovato sul nastro. Per fermare il caricamento premere <BREAK>. Analogamente, CLOADM e LOADM "CAS: hanno la stessa funzione. Normalmente è possibile "udire" la trasmissione dei dati programma dalla cassetta; ciò può essere evitato introducendo un comando SOUND OFF prima di quello di caricamento.

LOAD "RAM: è simile a LOAD "CAS: , ma RAM: può essere omesso.

I programmi possono anche essere caricati da linea di comunicazione, ma per questo occorre specificare le condizioni di ricezione attraverso i parametri di comunicazione (vedere a tale proposito il Capitolo 8).

LOG Funzione

LOG calcola il logaritmo naturale di un numero dato.

Formato: LOG(x)

dove: x è un'espressione numerica, che deve essere >0.

Esempio:
```
PRINT LOG(10)
 2.302585092994
Ok
```

LPOS Funzione

LPOS restituisce la posizione attuale della testina della stampante (nell'ambito del buffer di linea della stampante).

Formato: LPOS(x)

dove: x è un argomento fittizio, normalmente a 0.

Questa funzione non fornisce necessariamente la posizione effettiva della testina.

LPRINT , LPRINT USING Istruzioni

LPRINT e LPRINT USING sono usate per la stampa di dati. Il formato è lo stesso delle istruzioni PRINT e PRINT USING.

Se viene usato un microplotter invece di una stampante, l'istruzione LPRINT viene anche usata - insieme a CHR$ - per modificare la modalità del plotter. Per maggiori informazioni su queste ed altre istruzioni di stampa, vedere i comandi MICROPLOTTER.

MAXFILES Funzione

MAXFILES è usato per dichiarare il massimo numero di file che possono essere aperti, o per fornire il valore dichiarato in precedenza.

Formato: MAXFILES = n
MAXFILES

dove: n è un intero tra 0 e 15.

Se non viene utilizzata l'istruzione MAXFILES = n per definire il numero massimo, il sistema assume per default il valore 0.

MAXFILES fornirà il numero dichiarato con l'istruzione MAXFILES = n.

Esempio:

```
MAXFILES = 10
Ok
PRINT MAXFILES
 10
Ok
```

MAXRAM Funzione

MAXRAM fornisce il più alto indirizzo di memoria disponibile per BASIC.

Formato: MAXRAM

Questa funzione sarà utilizzata normalmente per definire il massimo valore possibile per il secondo argomento, nell'istruzione CLEAR.

MENU Comando

MENU provoca l'uscita dall'ambiente BASIC e il rientro al menù.

Formato: MENU

Anche il tasto funzione F8, premuto quando il sistema è in BASIC, provoca il ritorno al menù.

MERGE Comando

MERGE (fusione) intergra le linee di un programma memorizzato in formato ISO (o proveniente da linea di comunicazione) con il programma attualmente in memoria.

Formato:
```
MERGE "CAS: nome-file"
MERGE "RAM: nome-file"
MERGE "COM: cbpsx"
```

dove: nome-file è il nome col quale il file è stato memorizzato.

CAS integra un programma su cassetta.

RAM integra un programma in RAM.

COM integra un programma da linea di comunicazione.

Le linee del programma esterno o in RAM sono fuse ordinatamente con quelle del programma in memoria. Se vi è qualche numero di linea duplicato, la linea del programma precedentemente in memoria viene sostituita con quella del programma di fusione.

Quando l'operazione è completata, il sistema torna allo Stato Comandi ed il programma attualmemte in memoria è il risultato della fusione.

MICROPLOTTER Comandi

L'istruzione BASIC per stampare sul microplotter è LPRINT. Vi sono alcuni codici ISO che vengono considerati come istruzioni: per inviarli si deve usare la funzione BASIC CHR$.

Esempio: LPRINT CHR$(08)

invia al microplotter un codice ISO di ritorno di un passo (08). Se con la stessa forma si invia un carattere stampabile, esso verrà stampato.

Esempio: LPRINT CHR$(90)

invia un codice corrispondente alla lettera Z (90), che verrà stampata.

Il microplotter ha due modalità operative: modo Testo (in cui si entra automaticamente all'accensione) e modo Grafico. La modalità può essere cambiata da Testo a Grafico inviando un codice 18, oppure da Grafico a Testo con il codice 17. In modo Grafico sono disponibili un certo numero di comandi speciali, che vengono descritti di seguito. Essi devono essere inviati come una stringa (tra virgolette).

Esempio: LPRINT "A"

riporta il microplotter in modo Testo.

COMANDI IN MODO TESTO

CHR$(08) Ritorno di un passo: sposta la penna indietro di un carattere; è utile per sottolineare.

CHR$(11) Interlinea indietro: muove la carta indietro di una riga; è utile per la sovrascrittura.

CHR$(18) Seleziona il modo Grafico.

COMANDI IN MODO GRAFICO

Qualora nella descrizione siano specificate le coordinate x e y, x identifica l'asse orizzontale e y quello verticale. Si ricordi che la larghezza utile della carta è di solo 480 unità (passi elementari); il microplotter accetta la stessa gamma di valori per x e y, ma interpreta i valori di x superiori a 480 come uguali a 480.

CHR$(17) Seleziona il modo Testo.

A Formato: LPRINT "A"

Ritorna il microplotter dal modo Grafico al modo Testo e contemporaneamente sposta il carrellino portapenna al margine sinistro.

Esempio:

```
10 LPRINT "A"
20 LPRINT "A"
```

Se il microplotter è in modo Grafico, questi due comandi provocheranno il primo il ritorno in modo Testo, il secondo la stampa della lettera A

Cn

Formato: LPRINT "Cn"

dove: n è un valore fra 0 e 3.

Colour (selezione colore); provoca la sostituzione della penna con una delle altre tre disponibili. Se le penne sono montate correttamente, il colore è identificato da n:

0 = nero; 1 = blu; 2 = verde; 3 = rosso.

Esempio: LPRINT "C2" inserisce la penna colore verde.

Dx,y,...

Formato: LPRINT "Dx1,y1,x2,y2,..."

Draw (tracciare); traccia una linea dalla posizione attuale della penna al punto individuato da x1y1, quindi da quest'ultimo punto a x2y2 ecc. Si noti che le coordinate dei punti sono riferite all'origine (margine sinistro della prima linea), salvo che questa sia stata ridefinita col comando "I".

I valori di x e y devono essere compresi fra -999 e +999, ma ogni valore di x che comporti il superamento del valore assoluto 480 non potrà oltrepassare tale limite sul microplotter.

Esempio: LPRINT "0,100,100,100,100,0,0,0"

traccia un quadrato di lato pari a 100 passi.

H

Formato: LPRINT "H"

Home (origine); sposta il carrellino portapenna all' origine senza tracciare linee.

I

Formato: LPRINT "I"

Initialize (posizione iniziale); definisce la posizione attuale della penna come posizione iniziale o "origine".

Jx,y,... Formato: LPRINT "Jx1,y1,x2,y2,..."

(tracciare relativo); è simile al comando "D", ma le coordinate, anziché assolute, sono interpretate come relative all'ultima posizione raggiunta dalla penna. Viene cioè tracciata una linea dalla posizione attuale della penna ad un punto spostato relativamente ad essa di x1y1, da quest'ultimo punto ad un altro spostato di x2y2, ecc.

Esempio: LPRINT "J0,100,100,0,0,-100,-100,0"

traccia lo stesso quadrato dell'esempio "D", partendo questa volta dalla posizione attuale della penna.

Ln Formato: LPRINT "Ln"

dove: n è un valore fra 0 e 15

Line type (tipo di linea); definisce il tipo di linea, continua o tratteggiata, da utilizzare in modo Grafico. Nel caso di linea tratteggiata, il valore di n stabilisce la lunghezza del tratto:
n= 0 linea continua;
n= 1 linea tratteggiata con il tratto più corto;
n=15 linea tratteggiata con il tratto più lungo.

Il valore di default è n=0, cioè linea continua.

Esempio: LPRINT "L15"

definisce linee tratteggiate con i tratti più lunghi.

Mx,y Formato: LPRINT "Mx,y"

Move absolute (spostamento assoluto); sposta la penna alle coordinate assolute x,y (rispetto all'origine), senza tracciare linee.

I valori di x e y devono essere compresi fra -999 e +999, ma ogni valore di x che comporti il superamento del valore assoluto 480 non potrà oltrepassare tale limite sul microplotter.

N Formato: LPRINT "N"

(stampa carattere definito da utente); scrive il carattere definito col comando "U" nella posizione attuale della penna.

Pstringa

Formato: LPRINT "Pstringa"

dove: "stringa" è la sequenza da stampare.

Print (stampa); comanda la scrittura dei caratteri specificati in "stringa". Questo comando permette di stampare un testo senza abbandonare il modo Grafico.

Esempio: LPRINT "PAndamento annuale:"

provoca la stampa della frase Andamento annuale: a partire dalla posizione attuale della penna.

Qn

Formato: LPRINT "Qn"

dove: n è un numero fra 0 e 3

(orientamento stampa); specifica l'orientamento, nella scrittura delle stringhe definite dai comandi "P" successivi:

n=0 orizzontale, da sinistra verso destra;
n=1 verticale, dall'alto verso il basso;
n=2 orizzontale, da destra verso sinistra (capovolta);
n=3 verticale, dal basso verso l'alto.

Il valore di default all'accensione è n=0

Esempio: LPRINT "Q2"

determina la stampa delle successive stringhe "P" da destra a sinistra (capovolte).

Rx,y

Formato: LPRINT "Rx,y"

Relative move (spostamento relativo); sposta la penna, rispetto alla posizione attuale, dei valori (relativi) definiti da x,y senza tracciare linee.

I valori di x e y devono essere compresi fra -999 e +999, ma ogni valore di x che comporti il superamento del valore assoluto 480 non potrà oltrepassare tale limite sul microplotter.

Sn

Formato: LPRINT "Sn"

dove: n è un intero compreso fra 0 e 63.

Size (dimensione); definisce la dimensione dei caratteri stampati con i successivi comandi "P". La dimensione varia proporzionalmente al valore di n:

- con n=0 si ottiene il carattere più piccolo, con base pari a quattro punti elementari ed altezza pari a sei punti elementari (matrice 4x6);

- con n=63 si ottiene il carattere più grande, con base pari a 256 punti ed altezza pari a 384 punti.

I valori intermedi si ottengono con la formula:

"dimensione minima" * (1 + n)

E' utile rapportare i valori dimensionali dei caratteri alla capacità sulla riga di scrittura (orizzontale); tenendo conto della spaziatura intercarattere (due punti per la matrice minima), la capacitè può essere espressa con la formula:

$$\text{"capacità"} = \frac{480}{6 * (1+n)}$$

Il valore di default (n=1), corrisponde ad una densità di 40 caratteri per riga.

L'interlinea, cioè la distanza tra la base dei caratteri su due righe continue, è pari al doppio dell'altezza del carattere.

Esempio:
```
LPRINT "S0"
LPRINT "PEsempio di stampa!"
```

provoca la stampa della stringa Esempio di stampa! con una densità di 80 caratteri per riga. Se la riga viene stampata più volte, la distanza fra due righe successive è di 12 punti elementari (2 volte 6).

Up,x,y,...

Formato: LPRINT "Up,x1,y1,p,x2,y2, p,xn,y,,0,0,0"

dove: p definisce il tipo di movimento
(0 = muovi senza tracciare, 1 = traccia)

x,y sono coordinate comprese fra -7 e +7
(relative alla posizione attuale della penna)

n è un intero compreso fra 1 e 15

User character (carattere definito da utente); permette di definire un carattere speciale, disegnato nell'ambito di una matrice di 7x7 punti, con un massimo di 15 movimenti della penna. Una volta definito, il carattere potrà essere stampato più volte utilizzando il comando LPRINT "N".

Esempio:

```
LPRINT "U1,0,7,1,7,7,1,7,0,1,0,0,1,7,7,0,0,7,1,7,0,0,0,0"
```

Definisce un simbolo costituito da un quadrato di 7x7 punti, con entrambe le diagcnali tracciate.
Dopo questa definizione, e fino a quando il comando "U" non sarà utilizzato nuovamente per definire un altro carattere, il simbolo potrà essere scritto nella posizione attuale della penna digitando semplicemente LPRINT "N".

Xp,s,n

Formato: LPRINT "Xp,s,n"

dove: p identifica gli assi (0 = asse x, 1 = asse y)

s definisce il numero dei passi fra le tacche di graduazione dell'asse, e deve essere un intero fra -999 e +999

n definisce il numero delle tacche sull'asse, e deve essere un intero fra 1 e 255

aXis (tracciatura assi); permette di tracciare gli assi cartesiani per disegnare un grafico. L'origine degli assi è la posizione attuale della penna. Per l'asse x il prodotto s*n non deve essere superiore a 480, per rimanere all'interno dell'area utile del microplotter.

Esempio:

```
10 LPRINT "H"
20 LPRINT "X1,40,10
30 LPRINT "H"
40 LPRINT "X0,40,10
```

Questa sequenza disegna entrambi gli assi x ed y, iniziando dall'origine, per una distanza di 400 passi nella direzione positiva e con tacche di riferimento ogni 40 passi. Se s fosse stato uguale a -40 nella linea 40, l'asse y sarebbe stato nella direzione negativa, cioè verso il basso.

MID$ Funzione

MID$ fornisce una stringa ricavandola da un'altra.

Formato: MID$ (x$,i,j)

dove: x$ è un'espressione stringa

i e j sono interi tra 1 e 255

Dalla stringa x$ viene ricavata una stringa di j caratteri, a partire dall'i-esimo. Se j è omesso o se è maggiore dei caratteri restanti, i caratteri forniti sono tutti i caratteri dopo l'i-esimo. Se i è maggiore del numero di caratteri di x$, viene restituita la stringa nulla.

Esempio:

```
10 A$ = "Olivetti M10"
20 B$ = MID$(A$,7,4)
30 PRINT B$
RUN
ti M
Ok
```

MOD Funzione

MOD restituisce il resto di una divisione fra interi.

Formato: n MOD m

dove: n è il dividendo ed m il divisore, entrambi interi.

Ad esempio 10 MOD 3 restituisce il valore 1.

MOTOR ON/OFF Istruzioni

MOTOR ON e MOTOR OFF controllano da programma il motore di un registratore di nastro a cassetta.

Formato: MOTOR ON
MOTOR OFF

Queste istruzioni abilitano o inibiscono la funzione degli interruttori, in un registratore a cassetta. MOTOR OFF permette di controllare automaticamente l'azionamento del motore del registratore durante il trasferimento dei dati; MOTOR ON riabilita il controllo manuale.

NAME Comando

NAME è usato per ridefinire il nome di un file.

Formato: NAME "aaa.xx" AS "bbb.xx"

dove:

aaa è il vecchio nome del file

bbb è il nuovo nome del file

xx è il suffisso che deve rimanere identico.

Se il vecchio nome del file non esiste o se il nuovo nome esiste già, viene visualizzato un errore FC (chiamata funzione illegale). Se non vengono specificati i suffissi, sarà visualizzato un errore FF (file non trovato).

NEW Comando

NEW cancella il programma attualmente nella memoria di lavoro ed inizializza tutte le variabili.

Formato: NEW

Tutte le variabili numeriche vengono messe a 0 e le variabili stringa sono poste uguali alla stringa nulla. Lo spazio per le stringhe allocato da un CLEAR precedente non viene toccato.

ON...GOSUB Istruzione

ON...GOSUB provoca il salto a una subroutine.

Formato:

```
ON COM GOSUB n
ON TIME$ = "hh:mm:ss" GOSUB n
ON KEY GOSUB a, b, c ...
ON x GOSUB a, b, c ...
```

dove: a, b, c, n sono numeri di riga del programma

COM provoca un salto alla riga n, se vi sono dati nel buffer di linea

TIME$ = "hh:mm:ss" determina un istante in cui saltare alla linea n

KEY provoca un salto a una linea particolare, che dipende da quale tra gli otto tasti funzione viene premuto

x è un'espressione numerica, il cui valore provoca un salto a una delle righe elencate

L'istruzione ON COM GOSUB è usata in concomitanza con l'istruzione COM ON/OFF/STOP. Posto che sia stata eseguita in precedenza un'istruzione COM ON, il controllo del programma passa alla linea indicata se sono stati ricevuti dati dalla linea di comunicazione. Se invece è stato eseguito un COM OFF, viene disabilitato il passaggio alla subroutine. COM STOP inibisce la subroutine fino a che non viene eseguito il prossimo COM ON. Per altre informazioni vedere le istruzioni COM ON/OFF/STOP.

La variabile stringa TIME$ stabilisce un istante in cui saltare alla linea specificata. Quando avviene il salto, viene anche eseguito automaticamente un TIME$ STOP. L'istruzione RETURN della subroutine provoca automaticamente un TIME$ ON, a meno che in essa non sia stato incluso un TIME$ OFF. L'istruzione ON TIME$ ...GOSUB agisce solo se il programma è in esecuzione al tempo specificato.

Per l'istruzione ON KEY GOSUB possono essere specificate fino a otto righe programma, corrispondenti agli otto tasti funzione F1-F8. Quando avviene il salto, viene anche eseguito automaticamente un KEY(n) STOP. L'istruzione RETURN della subroutine provoca automaticamente un KEY(n) ON, a meno che in essa non sia stato incluso un KEY(n) OFF.

ON x GOSUB ... viene usata per saltare a righe diverse, a seconda del valore di x. Se x=1 il controllo viene ceduto alla prima riga elencata, se x=2 alla seconda e così via. Se x non è intero, la parte decimale viene scartata. Se x=0 o è maggiore del numero di righe elencate, il programma prosegue con l'istruzione successiva (purchè x < 255). Se x non è compreso nell'intervallo cui devono appartenere gli interi, sarà evidenziato l' errore FC (chiamata funzione illegale).

Se si verifica l'identificazione di un errore a seguito dell'istruzione ON ERROR, vengono disabilitati tutti gli accessi a subroutine.

Esempio:

```
10 INPUT "Scegli 1, 2 o 3";A
20 ON A GOSUB 100,200,300
30 GOTO 10
100 PRINT "subroutine 1": RETURN
200 PRINT "subroutine 2": RETURN
300 PRINT "subroutine 3": RETURN
RUN
Scegli 1, 2 o 3? 2      (l'utente introduce 2)
subroutine 2
Scegli 1, 2 o 3? C      (l'utente preme <CTRL> + C)
Break in 10
Ok
```

ON ... GOTO Istruzione

ON ... GOTO provoca un salto a un'altra riga di programma.

Formato:

```
ON ERROR GOTO n
ON x GOTO a, b, c ...
```

dove: a, b, c, n sono numeri di riga del programma

ERROR abilita il salto su errore, e specifica la prima riga della routine di gestione dell'errore (n).

x è un'espressione numerica il cui valore provoca il salto a una delle righe elencate.

Dopo un'istruzione ON ERROR GOTO ogni tipo di errore, compresi gli errori in Stato Comandi, ad es. SN (errore sintattico), provocherà un salto alla riga specificata per l'esecuzione della subroutine di gestione di errore. Normalmente un errore causerebbe l'interruzione del programma, ma una routine di trattamento errori permette al programma di recuperarlo e di proseguire. La routine di salto su errore può essere disabilitata eseguendo un'istruzione ON ERROR GOTO 0. Per tornare all'esecuzione normale del programma al termine della routine, utilizzare l'istruzione RESUME .

L'istruzione ON x GOTO viene utilizzata per saltare in punti diversi a seconda del valore di x, come per l'istruzione ON ... GOSUB. Se x=1, il controllo viene ceduto alla prima riga elencata, se x=2 alla seconda e così via. Se x non è intero, la parte decimale viene scartata. Se x=0 o è maggiore del numero di righe elencate, il programma prosegue con l'istruzione successiva (purchè x < 255). Se x non è compreso nell' intervallo cui devono appartenere gli interi, sarà evidenziato l' errore FC (chiamata funzione illegale).

OPEN Istruzione

OPEN è l'istruzione usata per aprire i file in RAM o su cassetta, o per assegnare un buffer a una periferica, trattandola come un file.

Formato:

```
OPEN "CAS:nome-file" FOR modo AS num-file
OPEN "RAM:nome-file" FOR modo AS num-file
OPEN "COM:cbpsx" FOR modo AS num-file
OPEN "LCD:" FOR OUTPUT AS num-file
OPEN "LPT:" FOR OUTPUT AS num-file
OPEN "WAND:" FOR INPUT AS num-file
```

dove:

CAS apre il registratore a cassette

RAM apre la RAM

COM apre la linea di comunicazione

LCD apre il display di M10

LPT apre la stampante

WAND apre il lettore di codici a barre

nome-file è il nome con cui il file è stato salvato

num-file è il numero assegnato al file, ed è preceduto dal simbolo £

modo può essere: INPUT, OUTPUT o APPEND.

Un'istruzione OPEN deve essere eseguita prima di ogni altra istruzione di input/output all'interno della quale sia richiesto un numero di file (num-file), ad esempio PRINT, PRINT USING, INPUT, LINE INPUT, INPUT$.

Num-file viene usato in altre istruzioni di input/output per riferirsi al file stesso, una volta aperto. Tale numero deve essere un intero compreso fra 1 ed il numero specificato dall'istruzione MAXFILES.

Le tre modalità, INPUT, OUTPUT e APPEND, sono valide per file in RAM. Per i file cassetta o linea di comunicazione sono utilizzabili solo INPUT e OUTPUT.

Un file non può essere aperto in una modalità se è già stato aperto in un'altra.

Quando la linea di comunicazione viene aperta come file, occorre specificare i parametri di comunicazione (vedere Capitolo 8, Fig.8-3).

OUT Istruzione

OUT è usata per inviare un byte dati ad una porta di output.

Formato: OUT n,i

dove: n è il numero della porta

i è il byte dati

Sia n che i sono interi, compresi tra 0 e 255. OUT è complementare alla funzione INP.

PEEK Funzione

PEEK restituisce il byte memorizzato all'indirizzo specificato.

Formato: PEEK (l)

dove: l è la locazione di memoria, che deve essere compresa tra -32768 e 65535.

Il valore fornito è decimale, compreso tra 0 e 255.

Se viene specificata un'allocazione negativa, essa viene sottratta da 65536, cioè -1 equivale a 65535.

PEEK è spesso usato per accedere ad informazioni memorizzate sia per mezzo dell'istruzione POKE che in subroutine in linguaggio macchina.

POKE Istruzione

POKE scrive un byte in uno specificato indirizzo di memoria.

Formato: POKE l,i

dove: l è la locazione di memoria, che deve essere compresa tra -32768 e 65535).

i è il byte da scrivere, che deve essere un intero compreso tra 0 e 255 (decimale).

Se viene specificata un'allocazione negativa, essa viene sottratta da 65536, cioè -1 equivale a 65535.

Utilizzare questa istruzione con cautela, dal momento che il sistema non controlla che la locazione di memoria specificata sia libera.

POS Funzione

POS fornisce la posizione attuale del cursore sul display.

Formato: POS(i)

dove: i è un argomento fittizio, normalmente 0.

Il margine sinistro del display è 0.

Esempio:
```
PRINT POS(0);POS(0);POS(0)
0  3  6
Ok
```

POWER Istruzione

Definisce il tempo prima dello spegnimento automatico.

Formato:
```
POWER i
POWER CONT
POWER OFF
POWER OFF, RESUME
```

dove: i è un intero tra 10 (1 minuto) e 255 (25.5 minuti).

Il sistema si spegne automaticamente se non vengono premuti tasti in un intervallo di tempo dichiarato nell'istruzione POWER i. La prestazione viene disabilitata dall'istruzione POWER CONT.

POWER OFF spegne il sistema; quando esso viene riacceso (posizionando l'interruttore principale prima in OFF poi in ON), verrà visualizzato il menù principale.

Se invece viene eseguita un'istruzione POWER OFF, RESUME , alla riaccensione manuale il sistema riprenderà lo stato in cui si trovava prima dell'esecuzione dell'istruzione stessa. Lo stesso avviene alla riaccensione manuale che segue uno spegnimento automatico.

PRESET Istruzione

PRESET è usato per specificare la posizione di un elemento sul display, che deve essere visualizzato in positivo o in negativo.

Formato: PRESET (x,y,c)

dove: x,y sono le coordinate dell'elemento; x deve essere compreso fra 0 e 239, y fra 0 e 63.

c è il parametro del colore; se c=0 l'elemento è nero (positivo), se c=1, oppure viene omesso, l'elemento è bianco (negativo).

Se le coordinate sono al di fuori dell'intervallo consentito, viene visualizzato un errore FC (chiamata funzione illegale).

Questa istruzione è simile alla PSET, ma in quest'ultima per c=1 l'elemento viene visualizzato in nero.

PRINT Istruzione

PRINT visualizza i dati sul display.

Formato: PRINT lista
? lista
PRINT n, lista

dove: lista è un elenco di espressioni, che devono essere separate o dalla virgola o dal punto e virgola.

? è un abbreviazione di PRINT.

n definisce la posizione di partenza della visualizzazione: per i dettagli vedere più avanti.

L'espressione può essere numerica o stringa, ma le stringhe devono essere racchiuse tra virgolette. Può essere specificato il nome di una variabile; in questo caso verrà visualizzato il contenuto dell' allocazione di memoria ad essa assegnata.

Il separatore punto e virgola fa sì che le espressioni siano visualizzate una di seguito all'altra, tranne che per le espressioni numeriche, che sono sempre seguite da uno spazio. I numeri positivi sono anche preceduti da uno spazio, che per i numeri negativi è occupato dal segno meno (-). Il separatore virgola visualizza i dati in forma tabellare. Gli arresti di tabulazione sono distanti 14 caratteri; sul display vengono formate due colonne, come si vede nell'esempio che segue.

Esempio:

```
Ok
PRINT 12;-345;"uno";" due";"cento"
 12 -345 uno duecento
Ok
a=2*4 : b$="Gen" : c=1983
PRINT a;b$;c
 8 Gen 1983
Ok
PRINT a,10,12,2*7,10+6,20-2
 8            10
12            14
16            18
Ok
```

Se al termine della lista vengono posti una virgola o un punto e virgola, la prossima istruzione PRINT inizierà la visualizzazione dalla posizione definita dal tipo di separatore.

Esempio:

```
10 PRINT 1;2;3;4;
20 PRINT 1;2,3,4
RUN
 1  2  3  4  1  2
 3            4
Ok
```

L'istruzione PRINT n è usata per iniziare la visualizzazione da una posizione definita; n rappresenta una posizione di carattere sul display. La prima posizione, indicata con 0, è l'angolo in alto a sinistra; le successive vengono contate procedendo da sinistra a destra e dall'alto verso il basso. Il valore più grande di n rappresenta pertanto l'angolo a destra in basso. Per il display LCD di M10 il valore massimo di n è 239. Per un display CRT esso è 999 nel caso di video da 40 caratteri per 25 righe, e 1990 nel caso di video da 80 caratteri per 25 righe.

PRINT USING Istruzione

PRINT USING visualizza i dati sul display con un formato definito.

Formato: PRINT USING "x"; lista

dove: x è un'espressione che definisce il formato dell'output; deve essere tra virgolette

lista è un elenco di espressioni, che devono essere separate dalla virgola o dal punto e virgola.

Non vengono inseriti spazi tra gli elementi da visualizzare, a meno che essi non siano inclusi nella stringa di formato.

Sono disponibili i seguenti comandi di formato:

COMANDO (per..)		DESCRIZIONE
!	(stringhe)	Specifica che deve essere visualizzato solo il primo carattere della stringa data.
ç ç	(stringhe)	Definisce la lunghezza della stringa da visualizzare, che è pari al numero degli spazi compresi fra i simboli, aumentato di 2.
£	(numeri)	Indica ogni posizione di cifra che dovrà essere riempita. I numeri con meno cifre di quelle richieste per la visualizzazione saranno preceduti da spazi. Si può inserire anche il separatore decimale, in qualunque posizione; se si specifica una cifra prima del separatore decimale, essa verrà sempre visualizzata, eventualmente con 0. I numeri con più cifre decimali di quelle richieste vengono arrotondati (arrotondamento matematico). I numeri maggiori del campo specificato sono visualizzati per intero, preceduti dal simbolo %; ciò vale anche per i numeri che dovessero superare il campo permesso a causa dell'arrotondamento.
+	(numeri)	Posto prima o dopo gli altri comandi di formato, specifica la posizione del segno rispetto al numero.
-	(numeri)	Posto dopo gli altri comandi di formato, specifica che i numeri negativi siano visualizzati con un segno meno finale.
**	(numeri)	Posto all'inizio della stringa di formato, definisce il riempimento con asterischi degli eventuali spazi iniziali. Il simbolo ** specifica anche due ulteriori posizioni di cifra.

$$ (numeri) Posto all'inizio della stringa di formato, fa sì che un segno di dollaro compaia alla immediata sinistra del numero. Il simbolo $$ specifica anche due ulteriori posizioni di cifra, una delle quali è occupata dal segno di $. Non è possibile usare questo comando con il formato esponenziale o con numeri negativi, a meno che il segno sia specificato come ultimo carattere della stringa (col comando + oppure -).

**$ (numeri) Combina gli effetti dei comandi ** e $$: gli spazi iniziali saranno occupati da asterischi ed un segno di $ verrà inserito immediatamente prima del numero. Questa combinazione specifica anche tre ulteriori posizioni di cifra, una delle quali è occupata dal segno di $.

, (numeri) Posto immediatamente a sinistra del separatore decimale, visualizza una virgola ogni tre cifre della parte intera del numero (per facilitarne la lettura). La virgola specifica una posizione di cifra, e non ha effetto sui numeri in formato esponenziale. Una virgola alla fine della stringa di formato viene visualizzata come un carattere.

^^^^ (numeri) Posto alla fine della stringa di formato, indica che il numero deve essere visualizzato in forma esponenziale (E+xx). Le cifre significative sono allineate a sinistra (la prima posizione potrà essere assegnata a spazio per il segno, in funzione degli altri comandi di formato).

Esempi:

```
A$ = "Olivetti" : B$ = "M1Ø"
Ok
PRINT USING "!";A$;" ";B$
O M
Ok

PRINT USING "Ç  Ç";A$;B$          (due spazi)
OlivM1Ø
Ok

A=123.45:B=6.789:C=-123.45:D=-6.789
Ok
PRINT USING "££.££ ";A;B;C;D
%123.45  6.79 %-123.45 -6.79
Ok

PRINT USING "£.£££^^^^- ";A;B;C;D
1.235E+Ø2  6.789E+ØØ  1.235E+Ø2- 6.789E+ØØ
Ok

PRINT USING "+**$£.£££ ";A;B;C;D
+$123.45Ø **+$6.789 -$123.45Ø **-$6.789
Ok
```

PRINT e PRINT USING servono anche a scrivere su file sequenziali.

Formato: PRINT num-file, dati
PRINT num-file, USING "x"; lista

dove: num-file è il numero con cui il file è stato aperto, preceduto dal simbolo £

dati può essere una lista di elementi, oppure di contenuti di variabili numeriche o stringa, separati da virgola o punto e virgola

"x" è un'espressione stringa che fornisce il formato dell'output

lista è una lista di espressioni, separate da virgole o da punto e virgola.

Queste istruzioni hanno lo stesso formato di PRINT e PRINT USING, ma in questo caso i dati sono trasferiti nel file prescelto. Utilizzando queste istruzioni si può scrivere un file su qualunque periferica, ma il file deve prima essere stato aperto in output. Vedere OPEN.

PSET Istruzione

PSET è usato per specificare la posizione di un elemento sul display, che deve essere visualizzato in positivo o in negativo.

Formato: PSET (x,y,c)

dove: x,y sono le coordinate dell'elemento; x deve essere compreso tra 0 e 239 e y tra 0 e 63.

c è il parametro del colore; se c=0, l'elemento è bianco (negativo), se c=1, oppure viene omesso, l'elemento è nero (positivo).

Se le coordinate sono esterne all'intervallo consentito, viene evidenziato un errore FC (chiamata funzione illegale).

Questa istruzione è simile alla PRESET, ma in quest'ultima per c=1 l'elemento viene visualizzato in bianco.

READ Istruzione

READ prende i valori da un'istruzione DATA e li assegna alle variabili.

Formato: READ var-1, var-2, ..., var-n

dove: var-1, var-2, var-n sono variabili numeriche o stringa

L'istruzione READ può accedere a una o più istruzioni DATA, che verranno lette in sequenza. Se si specificano più variabili dei dati disponibili sarà visualizzato l'errore OD (out of data = mancano dati). Le variabili devono corrispondere al tipo di dati memorizzati o verrà emesso un errore SN (sintassi errata).

Se vi sono meno variabili che dati, le istruzioni READ che seguono inizieranno a leggere dati dal primo non ancora letto. Se nel totale vi sono più dati che variabili al termine delle istruzioni READ, i dati non letti saranno ignorati.

L'istruzione RESTORE è usata per ripartire dal primo elemento della prima istruzione DATA.

Esempio:

```
Ok
10 READ A,B$,C,D$
20 PRINT A,B$,C,D$
30 DATA 56,AB,-3.55E+20,6abc7
RUN 10
56            AB
-3.55E+20     6abc7
Ok
```

REM Istruzione

REM permette di inserire in un programma dei commenti, che non ne influenzano l'esecuzione ma che appaiono nel listato.

Formato:

```
REM
'
```

L'apostrofo può essere utilizzato come abbreviazione di REM. Tutto il testo dopo REM viene considerato parte del commento; perciò, in una riga contenente più istruzioni, l'istruzione REM deve essere l'ultima.

E'anche possibile saltare a un'istruzione REM con un GOTO o un GOSUB; in questo caso il programma continuerà con l'istruzione eseguibile successiva.

REM non può essere usata come parte di un'istruzione DATA perchè non verrebbe riconosciuta, ma interpretata come dato.

RESTORE Istruzione

RESTORE viene utilizzata per rileggere le istruzioni DATA.

Formato: RESTORE n

dove: n è un numero di riga.

Dopo un'istruzione RESTORE, la READ successiva accederà al primo elemento dell'istruzione DATA sulla riga specificata. Se n è omesso, la READ accederà alla prima istruzione DATA del programma. Se la riga n non esiste, verrà evidenziato un errore UL (undefined line = linea non definita).

RESUME Istruzione

RESUME fa proseguire il programma dopo l'esecuzione di una subroutine di recupero di errore.

Formato:

```
RESUME
RESUME 0
RESUME NEXT
RESUME n
```

dove: n è un numero di riga.

RESUME e RESUME 0 provocano la continuazione dell'esecuzione a partire dall'istruzione che ha causato l'errore. RESUME NEXT provoca la continuazione a partire dall'istruzione successiva a quella che ha causato l'errore. Con RESUME n il programma continua dalla riga n.

Se si incontra una RESUME che non sia in una routine di errore viene visualizzato un errore RW (resume without error = resume senza errore).

RIGHT$ Funzione

RIGHT$ restituisce una stringa di caratteri posti nella parte destra di una stringa specificata.

Formato: RIGHT$(x$,i)

dove: x$ è una stringa

i è un intero tra 0 e 255.

Vengono forniti gli ultimi i caratteri della stringa x$. Se i è maggiore della lunghezza della stringa, viene restituita l'intera stringa; se i è zero, viene fornita una stringa nulla.

Esempio:

```
A$ = "Olivetti M10"
B$ = RIGHT$(A$,8)
PRINT B$
etti M10
Ok
```

RND Funzione

RND restituisce un numero pseudo-casuale compreso tra 0 e 1.

Formato: RND(x)

dove: x è un numero.

Se x>0 viene generata la stessa serie di numeri pseudo-casuali ogni volta che viene eseguito il programma. Se x<=0, viene ripetuto il numero generato in precedenza nella serie. Ciò è utile per il 'debugging'. La sequenza di numeri generata può essere 'disseminata' all'interno di un programma utilizzando la funzione TIME$.

Esempio:

```
10 J=VAL(RIGHT$(TIME$,2))
20 FOR I=1 TO J
30 RN=RND(1):NEXT
```

fornisce 60 valori diversi di RN (questo è solo uno dei tanti modi in cui si può distribuire la sequenza).

RUN e RUNM sono usati per caricare un programma nell'area di memoria di esecuzione programmi e per lanciarne l'esecuzione. I programmi non sono registrati nella directory, a meno che non sia stato usato in precedenza il comando SAVE.

Formato:

```
RUN n"nome.xx",R
RUN "CAS:nome-file",R
RUN "RAM:nome-file",R
RUN "COM:cbpsx",R
RUNM "CAS:nome-file"
RUNM "RAM:nome-file"
```

dove:

n è un numero riga

nome.xx è il nome del file con relativo suffisso

nome-file è il nome con cui il file è stato salvato

R mantiene aperti i file dati

CAS carica ed esegue un programma da cassetta

RAM carica ed esegue un programma da RAM

COM carica ed esegue un programma da linea

RUN carica in memoria ed esegue programmi BASIC, cancellando il programma eventualmente caricatovi in precedenza. RUNM carica ed esegue programmi in linguaggio macchina.

Un programma proveniente da RAM può essere eseguito a partire da una certa linea, specificata da n. Se non viene specificato il suffisso .xx , il sistema cercherà dapprima tra tutti i file con suffisso .BA , quindi tra quelli con suffisso .DO .

Di norma il comando RUN chiude tutti i file aperti. L'opzione R serve invece a mantenere aperti i file dati. RUN si comporta quindi come il comando LOAD... ,R.

SAVE , SAVEM Comandi

SAVE e SAVEM sono usati per memorizzare i programmi.

Formato:

```
SAVE "RAM:nome-file",A
SAVE "CAS:nome-file",A
SAVE "COM:cbpsx"
SAVEM "RAM:nome-file",a,b,c
SAVEM "CAS:nome-file",a,b,c
CSAVE "nome-file"
CSAVEM "nome-file",a,b,c
```

dove: RAM salva il programma attuale in RAM

CAS salva il programma attuale su cassetta

nome-file è il nome con cui il programma sarà riconosciuto

A è un'opzione con cui si specifica di salvare il programma in formato ISO

COM salva il programma attuale su linea di comunicazione, in formato ISO specifica un programma in linguaggio macchina

a,b,c sono indirizzi.

CSAVE e CSAVEM registrano programmi su cassetta in formato binario

SAVE "CAS: (senza l'opzione A) e CSAVE hanno lo stesso effetto; entrambi memorizzano programmi BASIC in formato binario su cassetta. Analogamente SAVEM "CAS: e CSAVEM per i programmi in linguaggio macchina.

In SAVE "RAM: e SAVEM "RAM: , si può omettere RAM: .

SAVE può essere usato anche per listare il programma attuale sul display, specificando: SAVE "LCD: per il display a cristalli liquidi, o SAVE "CRT: per un video esterno.

I programmi possono essere salvati su un altro sistema, via linea di comunicazione. Le condizioni di trasmissione e di ricezione devono essere compatibili, cioè l'apparecchiatura remota deve eseguire il caricamento utilizzando gli stessi parametri di comunicazione (cbpsx) specificati per SAVE (vedere Capitolo 8, Figura 8-3).

SCREEN definisce il tipo di display utilizzato.

Formato: SCREEN d,k

dove: d specifica il display; se d=0 è il display LCD, se d=1 è un video CRT

k controlla la visualizzazione del significato dei tasti funzione; se k=0 è disabilitata, se k=1 è abilitata.

Il valore di default di entrambi i parametri è 0. Qualora d sia posto uguale a 1 e non risulti connesso nessun CRT, viene emesso un errore FC chiamata funzione illegale).

SGN Funzione

SGN fornisce il segno di un numero.

Formato: SGN(x)

La funzione ritorna 1 se x<0, 0 se x=0 e -1 se x>0.

SIN Funzione

SIN fornisce il seno di un angolo.

Formato: SIN (x)

dove: x è un angolo espresso in radianti

Esempio:

```
Ok
PRINT SIN(0.5)
 .47942553860422
Ok
```

SOUND **Istruzione**

SOUND emette una nota specificata.

Formato: SOUND nota,lunghezza
SOUND ON
SOUND OFF

dove: nota è un intero compreso tra 0 e 16383; la tabella sotto riportata elenca i valori per particolari note

lunghezza è un intero compreso tra 0 e 255; per un intero i la lunghezza della nota è (i+1)*20 ms.

SOUND ON abilita l'altoparlante e SOUND OFF lo disabilita.

SOUND ON/OFF è anche usato per abilitare o disabilitare l'altoparlante durante il caricamento programmi da cassetta o l'utilizzo di TELCOM.

OTTAVE →	1	2	3	4	5	6
NOTE						
DO		9394	4697	2348	1171	587
DO#		8866	4433	2216	1103	554
RE		8368	4184	2092	1043	523
RE#	15800	7900	3950	1975	987	493
MI	14912	7456	3728	1864	932	466
FA	14064	7032	3516	1758	879	439
FA#	13284	6642	3321	1660	830	415
SOL	12538	6269	3134	1567	783	
SOL#	11836	5918	2954	1479	739	
LA	11172	5586	2793	1396	693	
LA#	10544	5272	2636	1318	659	
SI	9952	4968	2484	1244	622	

SPACE$ restituisce una stringa di spazii.

Formato: SPACE$(x)

dove: x è un numero tra 0 e 255.

Se x non è un intero, la parte dopo la virgola decimale viene ignorata.

Esempio:

```
Ok
A$ = SPACE$(5)
Ok
PRINT "x";A$;"y";A$;"z"
x     y     z
Ok
```


SQR calcola la radice quadrata di un numero.

Formato: SQR(x)

dove: x è un numero positivo.

Se x è negativo, viene visualizzato un errore FC (chiamata funzione illegale).

Esempio:

```
Ok
PRINT SQR(50)
 7.0710678118655
Ok
```

STOP Istruzione

STOP arresta l'esecuzione del programma e provoca il ritorno in Stato Comandi.

Formato: STOP

Questa istruzione può essere inserita in qualsiasi punto del programma; quando viene eseguita, viene emesso il messaggio:

Break in n (interruzione in n)

dove: n è il numero riga.

I file vengono lasciati aperti e l'esecuzione può essere ripresa con il comando CONT, purchè il programma non venga modificato durante l'arresto.

STRING$ Funzione

STRING$ crea una stringa, di una lunghezza specificata, i cui caratteri sono tutti uguali ad uno specificato.

Formato:
```
STRING$(a,b)
STRING$(a,x$)
```

dove: a,b sono interi, compresi tra 0 e 255

x$ è una stringa

La stringa che viene creata è costituita da a caratteri, tutti uguali al codice ISO b oppure al primo carattere della stringa x$.

Esempio:
```
Ok
A$ = STRING$(20,47)
Ok
PRINT A$
////////////////////
Ok
```

STR$ Funzione

STR$ converte il valore di un'espressione numerica in forma di stringa.

Formato: STR$(x)

dove: x è un numero

La stringa fornita è utilizzata per trasferire dati alle periferiche.

Esempio:

```
Ok
PRINT STR$(34)
 34
Ok
```

Il numero visualizzato (34) è memorizzato come stringa di due caratteri, cioè il carattere ISO 3 seguito dal carattere ISO 4.

Nota: la funzione inversa è eseguita da VAL.

TAB Funzione

TAB è usato nelle istruzioni PRINT e LPRINT per posizionarsi in un punto desiderato di una linea.

Formato: TAB(i)

dove: i è un intero tra 0 e 255.

Se il valore di i è inferiore alla posizione attuale del cursore, il prossimo elemento verrà visualizzato o stampato immediatamente dopo il cursore (con gli spazi opportuni se si tratta di un numero).

i=0 rappresenta il margine sinistro del display; cifre crescenti si riferiscono a posizioni crescenti verso destra (40 è il primo carattere della seconda riga, ... 255 è il sedicesimo carattere della settima riga).

Esempio:

```
Ok
PRINT "a" TAB(5) "b" TAB(3) "c" TAB(50) "d"
a    bc
                d
Ok
```

TAN Funzione

TAN calcola la tangente di un angolo.

Formato: TAN(x)

dove: x è un angolo espresso in radianti.

Esempio:

```
Ok
PRINT TAN(0.5)
 .54630248984381
Ok
```

TIME$ Istruzione e funzione

TIME$ definisce l'ora attuale per l'orologio interno, oppure restituisce l'ora attuale. TIME$ ON/OFF/STOP sono invece istruzioni usate per abilitare o inibire l'aggancio ad una subroutine per mezzo dell' istruzione ON TIME$ ... GOSUB.

Formato:

```
TIME$ = "hh:mm:ss"
TIME$
TIME$ ON
TIME$ OFF
TIME$ STOP
```

dove: "hh:mm:ss" è il formato usato per definire l'ora attuale.

Quando si imposta il tempo, per hh si devono usare valori tra 00 e 23. SI NOTI CHE I VALORI COMPRESI TRA 24 E 29 SARANNO ACCETTATI DAL SISTEMA, MA SARANNO MEMORIZZATI COME 00, MENTRE VALORI MAGGIORI DI 29 DARANNO LUOGO ALLA SEGNALAZIONE DI ERRORE SN (sintassi errata).

TIME$ ON abilita l'aggancio a una subroutine e TIME$ OFF lo disabilita. TIME$ STOP inibisce l'aggancio fino a che non venga eseguita un'istruzione TIME$ ON: se l'ora indicata da ON TIME$ ... GOSUB è già passata, viene eseguito un salto immediato alla subroutine.

Se TIME$ viene usato come funzione, viene fornita l'ora attuale.

Esempio:

```
Ok
PRINT TIME$
16:13:16
Ok
```

VAL Funzione

VAL converte la rappresentazione stringa di un numero nel suo valore numerico.

Formato: VAL(x$)

dove: x$ è una stringa.

Spazi iniziali, tabulazioni e interlinee vengono ignorati.

Esempio:

```
PRINT VAL(" 34")
 34
Ok
```

La funzione inversa è eseguita da STR.

VARPTR Funzione

VARPTR fornisce l'indirizzo del primo byte di una variabile o un file.

Formato: VARPTR(x)
VARPTR(n)

dove: x è una variabile di qualsiasi tipo

n è il numero con il quale il file è stato aperto, preceduto dal simbolo £.

Questa funzione viene spesso utilizzata per ottenere l'indirizzo di una variabile o di una matrice, per poterlo inviare ad una subroutine in linguaggio macchina.

A ogni variabile chiamata attraverso VARPTR deve essere già stato assegnato un valore, altrimenti viene evidenziato l'errore FC (chiamata funzione illegale).

Il numero fornito sarà compreso tra -32768 e 32767. Se l'indirizzo ottenuto è negativo, si deve aggiungere 65536 per ottenere l'indirizzo effettivo.

Per le matrici, devono essere stati assegnati in precedenza dei valori a tutte le variabili elementari; per ottenere l'indirizzo del primo elemento della matrice si usa la forma VARPTR (A(0)).

Per un file, la funzione fornisce l'indirizzo di partenza del blocco di controllo.

Esempio

```
Ok
10 OPEN "LCD:" FOR OUTPUT AS £1
20 PRINT VARPRT(£1)
RUN
-4755
Ok
```

WIDTH Istruzione

WIDTH seleziona la larghezza del video CRT, a 40 od 80 caratteri.

Formato: WIDTH n

dove: n è 40 o 80 a seconda della larghezza voluta.

A. SOMMARIO DELLE SPECIFICHE TECNICHE

Dimensioni

300mm x 220mm x 60mm.

Peso

1800 grammi.

Alimentazione

6V corrente continua, sia utilizzando 4 pile alcaline a secco da 1.5V - dimensione AA -, che attraverso un adattatore collegato alla rete di alimentazione pubblica.

Microprocessore

80C85 (CPU a 8 bit).

RAM

8K byte standard, espandibile a 32K byte a moduli di 8K.

ROM

32K byte standard; possibilità di inserimento di una ROM addizionale contenente funzioni aggiuntive (altri 32K byte).

B. MESSAGGI D'ERRORE BASIC

Gli errori su M10 vengono segnalati con codici alfabetici, elencati nel seguito. I numeri indicati a fianco di ciascun codice di errore sono degli interi, riconoscibili dal BASIC, che devono essere utilizzati con l'istruzione ERROR. Per facilitare la ricerca, gli errori sono riportati anche in ordine numerico (a fine appendice).

Sono disponibili 255 codici d'errore, ma non tutti sono attualmente utilizzati dal BASIC. Alcuni numeri d'errore, fra cui i numeri riservati per uso futuro, provocano una segnalazione di codice d'errore non stampabile (UE).

ERRORE CODICE	NUM	DESCRIZIONE
AO	53	File già aperto; è stata emessa una OPEN in modalità sequenziale per un file già aperto; oppure si è tentato di eseguire un comando KILL su un file aperto.
BS	9	Indice fuori dai limiti; si è fatto riferimento a un elemento di matrice o con un indice fuori dalle dimensioni della matrice, o con un numero sbagliato di indici.
BN	51	Errato numero di file; un'istruzione o un comando fa riferimento a un file il cui numero non appartiene all'intervallo specificato con MAXFILE, oppure il file corrispondente non è aperto.
CF	58	File non aperto; il file specificato in un'istruzione PRINT, INPUT o simili non è stato aperto.
CN	17	Non possibile continuare; si è cercato di riprendere l'esecuzione di un programma: a. che si è bloccato per un errore; b. che è stato modificato durante una pausa nell'-esecuzione (ottenibile con <CTRL> + C); c. che non esiste in memoria.
DD	10	Vettore ridimensionato; due o più istruzioni DIM fanno riferimento alla stessa matrice, oppure viene eseguita un'istruzione DIM dopo che per una data matrice erano state assegnate le dimensioni di default.
DS	56	Istruzione immediata in un file; durante il caricamento di un file in formato ISO è stata incontrata una

		istruzione immediata (diretta) e il caricamento si è bloccato.
EF	54	Input oltre la fine del file; si è tentato di eseguire un'istruzione di input su un file già completato oppure su un file vuoto. Usare la funzione EOF per sondare la fine del file ed evitare questo errore.
FC	5	Chiamata funzione illegale; viene passato un argomento fuori dai limiti prescritti ad una funzione numerica o stringa.
		Un errore di questo tipo può succedere, anche quando: a. un indice di matrici risulti negativo o eccezzionalmente grande; b. una funzione LOG abbia un argomento negativo o nullo; c. una funzione SQR abbia un argomento negativo; d. è stato dato un argomento non corretto a una delle seguenti istruzioni:INP, INSTR, LEFT$, MID$, ON ... GOTO, OUT, PEEK, POKE, RIGHT$, SPACE$ o STRING$.
FF	52	File non trovato; un'istruzione LOAD, KILL o OPEN fanno riferimento a un file che non esiste in memoria.
FL	57	Troppi file; è stato tentato di creare un nuovo file, utilizzando un comando SAVE o una OPEN, quando tutte le directory sono piene.
ID	12	Comando immediato illegale; è stata introdotta come linea ad esecuzione immediata una istruzione non consentita in Stato Comandi.
IE	50	Errore interno; si è verificata un'anomalia interna. Segnalare le condizioni d'errore all'Organizzazione di Assistenza.
IO	23	Errore di input/output; si è verificato un errore di input/output durante operazioni su cassetta, stampante o video. Si tratta di un errore non recuperabile da BASIC.
LS	15	Stringa troppo lunga; si è tentato di creare una stringa con più di 255 caratteri.
MO	22	Operando mancante; una espressione contiene un operatore non seguito da operando.
NF	1	NEXT senza FOR; una variabile in un'istruzione NEXT non corrisponde ad alcuna variabile di una istruzione FOR eseguita in precedenza.
NM	55	Nome file errato; è stato usato un formato non consentito per specificare un nome di file, in uno dei comandi LOAD, SAVE, KILL, NAME, etc.

NR	19	Manca l'istruzione RESUME; il controllo è stato passato a una routine di gestione degli errori che non contiene l'istruzione RESUME.
OD	4	Dati insufficienti; viene eseguita un'istruzione READ quando nel programma non sono rimaste istruzioni DATA con dati da leggere.
OM	7	Memoria insufficiente; un programma è troppo grande, ha troppi cicli o subroutine, o troppe variabili, o espressioni troppo complesse.
OS	14	Spazio per stringhe insufficiente; le variabili stringa hanno provocato il superamento della memoria libera rimanente (il BASIC alloca lo spazio per le stringhe in modo dinamico, finché non succede un errore di questo tipo).
OV	6	Overflow; il risultato di un calcolo è troppo grande per essere rappresentato con il formato BASIC di rappresentazione dei numeri.
RG	3	RETURN senza GOSUB; viene incontrata un'istruzione RETURN alla quale non corrisponde alcuna precedente istruzione GOSUB.
RW	20	RESUME senza routine d'errore; viene incontrata un'-istruzione RESUME senza prima essere entrati in una routine di gestione degli errori.
SN	2	Errore sintattico; una riga programma contiene una sequenza non corretta di caratteri, ad esempio parentesi non accoppiate, parole chiave con errori ortografici, punteggiatura non corretta.
ST	16	Espressione stringa troppo complessa; un'espressione stringa è troppo lunga o troppo complessa. E' opportuno dividerla in più espressioni semplici.
TM	13	Tipo di variabile errato; un nome di una variabile stringa è stato associato ad un valore numerico o viceversa. Oppure una funzione che richiede argomento numerico ha ricevuto un argomento stringa o viceversa.
UE	21	Errore non stampabile; non è disponibile un messaggio per la condizione d'errore trovata.
UE	24-29	Errore non stampabile; questi codici non sono stati definiti, ma sono riservati ad utilizzi futuri del BASIC.
UE	59-255	Errore non stampabile; questi codici non sono definiti, e possono essere utilizzati da programmi utente.
UL	8	Linea non definita; un'istruzione DELETE, GOTO, GOSUB o IF ... ELSE fa riferimento a una linea inesistente.

/0	11	Divisione per zero; viene incontrata una divisione per zero, oppure si ha un elevamento a potenza dove il valore della base è zero e l'esponente è negativo.

La tabella che segue elenca i codici d'errore in ordine numerico.

1	NF	NEXT senza FOR
2	SN	Errore sintattico
3	RG	RETURN senza GOSUB
4	OD	Dati insufficienti
5	FC	Chiamata funzione illegale
6	OV	Overflow
7	OM	Memoria insufficiente
8	UL	Linea non definita
9	BS	Indice fuori dai limiti
10	DD	Vettore ridimensionato
11	/0	Divisione per zero
12	ID	Comando immediato illegale
13	TM	Tipo di variabile errato
14	OS	Spazio per stringhe insufficiente
15	LS	Stringa troppo lunga
16	ST	Espressione stringa troppo complessa
17	CN	Non possibile continuare
19	NR	Manca l'istruzione RESUME
20	RW	RESUME senza routine d'errore
21	UE	Errore non stampabile
22	MO	Operando mancante
23	IO	Errore di input/output
24-49	UE	Errore non stampabile
50	IE	Errore interno
51	BN	Errato numero di file
52	FF	File non trovato
53	AO	File già aperto
54	EF	Input oltre la fine del file
55	NM	Nome file errato
56	DS	Istruzione immediata in un file
57	FL	Troppi file
58	CF	File non aperto
59-255	UE	Errore non stampabile

C. TABELLE DI EQUIVALENZA DEI CODICI

Le tabelle che seguono riportano i codici ASCII utilizzati da M10, sia come caratteri di controllo che come caratteri grafici. La prima tabella comprende i 32 caratteri di controllo, che possono assumere funzioni diverse a seconda che M10 si trovi in modalità BASIC, TEXT o TELCOM.

CARATTERI DI CONTROLLO

Le funzioni indicate nella colonna BASIC con un asterisco, possono essere attivate direttamente dalla tastiera. Le altre possono essere attivate da BASIC, utilizzando il formato

PRINT CHR$(n)

dove n è il numero del codice ASCII (valore decimale).

Esempio: nella tabella BREAK è contrassegnato con asterisco, mentre Cicalino non lo è. Di conseguenza

PRINT CHR$(07) seguito da <ENTER> provoca una segnalazione acustica (beep);

<CTRL> + C in tastiera interrompe un programma (break).

COD. ASCII	EQUIVALENTE IN TASTIERA	FUNZIONE IN: BASIC	TEXT	TELCOM
00	<CTRL> + @	nessuna	nessuna	nessuna
01	<CTRL> + A	nessuna	Curs. a inizio parola precedente	nessuna
02	<CTRL> + B	nessuna	Curs. a ultima riga del display	nessuna
03	<CTRL> + C	Break *	Annulla Find, Load, Save, Select, Print	nessuna
04	<CTRL> + D	nessuna	Curs. a inizio parola seguente	nessuna
05	<CTRL> + E	nessuna	Curs. una riga in alto	nessuna
06	<CTRL> + F	nessuna	Curs. uno spazio a destra	nessuna
07	<CTRL> + G	Cicalino	Save (salva)	Cicalino
08	<CTRL> + H	Cancella carattere	Cancella carattere	Cancella carattere
09	<CTRL> + I	Tabula *	Tabula	Tabula
10	<CTRL> + J	Interlinea	nessuna	Interlinea
11	<CTRL> + K	nessuna	nessuna	Interlinea
12	<CTRL> + L	Cancella schermo	Select (scegli)	Cancella schermo
13	<CTRL> + M	Ritorno a capo	Ritorno a capo	Ritorno a capo
14	<CTRL> + N	nessuna	Find (trova)	nessuna
15	<CTRL> + O	nessuna	Copy (copia)	nessuna
16	<CTRL> + P	nessuna	P (se premuto due volte)	nessuna
17	<CTRL> + Q	nessuna	Curs. a inizio riga	XON (linea abilitata)
18	<CTRL> + R	nessuna	Curs. a fine riga	nessuna
19	<CTRL> + S	Pause *	Curs. uno spazio a sinistra	XOFF (linea disabilitata)

COD. ASCII	EQUIVALENTE IN TASTIERA	FUNZIONE IN: BASIC	TEXT	TELCOM
20	<CTRL> + T	nessuna	Curs. a prima riga del display	nessuna
21	<CTRL> + U	Cancella linea *	Cut (taglia)	Cancella riga attuale
22	<CTRL> + V	nessuna	Load (carica)	nessuna
23	<CTRL> + W	nessuna	Curs. a inizio file	nessuna
24	<CTRL> + X	Cancella linea *	Curs. una riga in basso	Cancella riga attuale
25	<CTRL> + Y	nessuna	Stampa il file	nessuna
26	<CTRL> + Z	nessuna	Curs. a fine file	nessuna
27	<CTRL> + 0	Escape	Salva il file e vai al Menù (se premuto due volte)	Escape
28	< ← >	nessuna	Curs. uno spazio a destra	Curs. uno spazio a destra
29	< → >	Cancella carattere	Curs. uno spazio a sinistra	Curs. uno spazio a sinistra
30	< ↑ >	nessuna	Curs. una riga in alto	Curs. una riga in alto
31	< ↓ >	nessuna	Curs. una riga in basso	Curs. una riga in basso

CARATTERI GRAFICI

COD. ASCII	CARATTERE GRAFICO	CARATTERE IN TASTIERA	COD. ASCII	CARATTERE GRAFICO	CARATTERE IN TASTIERA
32	Spazio	<SPAZIO>	57	9	9
33	!	!	58	:	:
34	"	"	59	;	;
35	£	£	60	<	<
36	$	$	61	=	=
37	%	%	62	>	>
38	&	&	63	?	?
39	'	'	64	§	§
40	(	(	65	A	<SHIFT> + A
41	)	)	66	B	<SHIFT> + B
42	*	*	67	C	<SHIFT> + C
43	+	+	68	D	<SHIFT> + D
44	,	,	69	E	<SHIFT> + E
45	-	-	70	F	<SHIFT> + F
46	.	.	71	G	<SHIFT> + G
47	/	/	72	H	<SHIFT> + H
48	0	0	73	I	<SHIFT> + I
49	1	1	74	J	<SHIFT> + J
50	2	2	75	K	<SHIFT> + K
51	3	3	76	L	<SHIFT> + L
52	4	4	77	M	<SHIFT> + M
53	5	5	78	N	<SHIFT> + N
54	6	6	79	O	<SHIFT> + O
55	7	7	80	P	<SHIFT> + P
56	8	8	81	Q	<SHIFT> + Q

COD. ASCII	CARATTERE GRAFICO	CARATTERE IN TASTIERA	COD. ASCII	CARATTERE GRAFICO	CARATTERE IN TASTIERA
82	R	<SHIFT> + R	107	k	K
83	S	<SHIFT> + S	108	l	L
84	T	<SHIFT> + T	109	m	M
85	U	<SHIFT> + U	110	n	N
86	V	<SHIFT> + V	111	o	O
87	W	<SHIFT> + W	112	p	P
88	X	<SHIFT> + X	113	q	Q
89	Y	<SHIFT> + Y	114	r	R
90	Z	<SHIFT> + Z	115	s	S
91	°	°	116	t	T
92	ç	ç	117	u	U
93	é	é	118	v	V
94	^	^	119	w	W
95	—	—	120	x	X
96	ù	ù	121	y	Y
97	a	A	122	z	Z
98	b	B	123	à	à
99	c	C	124	ò	ò
100	d	D	125	è	è
101	e	E	126	ì	ì
102	f	F	127-159	(non usato)	
103	g	G	160	◄	<ENTER>
104	h	H	161	Ä	<G + S> + &
105	i	I	162	Ö	<G + S> + 1
106	j	J	163	Ü	<G + S> + 2

COD. ASCII	CARATTERE GRAFICO	CARATTERE IN TASTIERA	COD. ASCII	CARATTERE GRAFICO	CARATTERE IN TASTIERA
164	Å	<G + S> + 3	189	³	<G + S> + F
165	Æ	<G + S> + 4	190	∫	<G + S> + G
166	Ø	<G + S> + 5	191	#	<G + S> + H
167	Ã	<G + S> + 6	192	≠	<G + S> + J
168	Õ	<G + S> + 7	193	≤	<G + S> + K
169	Ç	<G + S> + 8	194	±	<G + S> + L
170	Ñ	<G + S> + 9	195	¼	<G + S> + M
171	←	<G + S> + 0	196	½	<G + S> + %
172	→	<G + S> + °	197	☎	<G + S> + >
173	↑	<G + S> + +	198	🏠	<G + S> + W
174	↓	<G + S> + §	199	[x	<G + S> + X
175	△	<G + S> + Q	200	±	<G + S> + C
176	µ	<G + S> + Z	201	♯	<G + S> + V
177	π	<G + S> + E	202	£	<G + S> + B
178	Σ	<G + S> + R	203	¥	<G + S> + N
179	Ω	<G + S> + T	204	ø	<G + S> + ?
180	√	<G + S> + Y	205	¤	<G + S> + .
181	≤	<G + S> + U	206	@	<G + S> + /
182	≥	<G + S> + I	207	°	<G + S> + !
183	∞	<G + S> + O	208	□	<GRPH> + ò
184	[	<G + S> + P	209	ä	<GRPH> + $
185	]	<G + S> + =	210	ö	<GRPH> + £
186	×	<G + S> + A	211	ü	<GRPH> + é
187	÷	<G + S> + S	212	à	<GRPH> + "
188	²	<G + S> + D	213	æ	<GRPH> + '

COD. ASCII	CARATTERE GRAFICO	CARATTERE IN TASTIERA	COD. ASCII	CARATTERE GRAFICO	CARATTERE IN TASTIERA
214	ø	<GRPH> + (	235	ô	<GRPH> + S
215	ã	<GRPH> + _	236	ù	<GRPH> + D
216	õ	<GRPH> + è	237	û	<GRPH> + F
217	ç	<GRPH> + ^	238	§	<GRPH> + G
218	ñ	<GRPH> + ç	239	|	<GRPH> + H
219	ì	<GRPH> + à	240	\	<GRPH> + J
220	¿	<GRPH> +)	241	º	<GRPH> + K
221	ij	<GRPH> + -	242	ª	<GRPH> + L
222	ß	<GRPH> + *	243	■	<GRPH> + M
223	à	<GRPH> + Q	244	▨	<GRPH> + ù
224	â	<GRPH> + Z	245	┌	<GRPH> + <
225	é	<GRPH> + E	246	┐	<GRPH> + W
226	è	<GRPH> + R	247	└	<GRPH> + X
227	ê	<GRPH> + T	248	┘	<GRPH> + C
228	ë	<GRPH> + Y	249	│	<GRPH> + V
229	ì	<GRPH> + U	250	─	<GRPH> + B
230	î	<GRPH> + I	251	|	<GRPH> + N
231	ï	<GRPH> + O	252	—	<GRPH> + ,
232	{	<GRPH> + P	253	┌	<GRPH> + ;
233	}	<GRPH> + ì	254	┘	<GRPH> + :
234	ò	<GRPH> + A	255	(non usato)	

AVVERTENZA: per ragioni di spazio, in questa tabella si è modificata la notazione standard; <G + S> equivale a <GRPH + SHIFT>

TABELLA DI CODICE A 8 BIT

M10

b_8	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
b_7	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
b_6	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
b_5	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
b_4 b_3 b_2 b_1	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
0 0 0 0	00	NUL	.	SP	0	§	P	ù	p			◀	µ	≠	□	â	\
0 0 0 1	01		DC1	!	1	A	Q	a	q			Ä	π	≤	ä	é	º
0 0 1 0	02			"	2	B	R	b	r			Ö	Σ	±	ö	è	ª
0 0 1 1	03	ETX	DC3	£	3	C	S	c	s			Ü	Ω	¼	ü	ê	■
0 1 0 0	04			$	4	D	T	d	t			Å	√	½	å	ë	▨
0 1 0 1	05		NAK	%	5	E	U	e	u			Æ	≤	☎	æ	ì	┌
0 1 1 0	06			&	6	F	V	f	v			Ø	≥	⌂	ø	î	┐
0 1 1 1	07	BEL		'	7	G	W	g	w			Ã	∞	[x	ã	ï	└
1 0 0 0	08	BS	CAN	(	8	H	X	h	x			Õ	[	±	õ	{	┘
1 0 0 1	09	HT		)	9	I	Y	i	y			Ç	]	⌗	ç	}	│
1 0 1 0	10	LF		*	:	J	Z	j	z			Ñ	×	£	ñ	ò	─
1 0 1 1	11		ESC	+	;	K	°	k	à			←	÷	¥	¡	ô	|
1 1 0 0	12	FF	FS	,	<	L	Ç	l	ò			→	²	¢	¿	ù	_
1 1 0 1	13	CR	GS	–	=	M	é	m	è			↑	³	¤	ij	û	┌┐
1 1 1 0	14		RS	.	>	N	^	n	ì			↓	∫	@	ß	§	└┘
1 1 1 1	15		US	/	?	O	_	o				△	#	°	à	|	

D. SEQUENZE DI ESCAPE IN BASIC E IN MODO TELCOM

Il codice ISO per Escape (ESC) è 27; usandolo in combinazione con altri caratteri, si possono generare un certo numero di sequenze (dette sequenze di Escape), valide sia in BASIC che in TELCOM.

In TEXT il carattere ESC è ottenuto con <CTRL> + 0; se premuto due volte, salva il file attuale e riporta al menù principale. Le altre sequenze non sono utilizzabili in modalità TEXT.

In BASIC e TELCOM, le sequenze di Escape non sono generabili da tastiera; dovrà invece essere utilizzato un formato del tipo:

PRINT CHR$(27)+"c" oppure PRINT CHR$(27)+CHR$(n)

dove c è il secondo carattere della sequenza e n è il codice ISO (decimale) per quello stesso carattere.

Si possono usare indifferentemente l'uno o l'altro formato per ottenere le funzioni elencate nella tabella che segue.

Inoltre sono validi anche i formati alternativi:

PRINT CHR$(27);"c" oppure PRINT CHR$(27);CHR$(n)

E' importante notare che alcune lettere sono utilizzate, con significati diversi, nella rappresentazione minuscola e maiuscola; esse hanno infatti codici ISO diversi.

SEQUENZA	FUNZIONE
ESC + E	Cancella il display
ESC + j	Cancella il display
ESC + K	Cancella fino a fine riga
ESC + J	Cancella fino a fine pagina
ESC + l	Cancella una riga intera
ESC + L	Inserisce una riga vuota
ESC + M	Cancella una riga
ESC + A	Sposta il cursore sulla riga precedente

SEQUENZA	FUNZIONE
ESC + B	Sposta il cursore sulla riga successiva
ESC + C	Sposta il cursore uno spazio a destra
ESC + D	Sposta il cursore uno spazio a sinistra
ESC + H	Sposta il cursore alla posizione "home" (0,0)
ESC + p	Pone il display in modo "negativo"
ESC + q	Annulla il modo "negativo" sul display
ESC + T	Protegge la riga 7 del display (che riporta il significato dei tasti funzione)
ESC + U	Toglie la protezione dalla riga 7 del display
ESC + W	Abilita lo scorrimento sul display
ESC + V	Annulla lo scorrimento sul display
ESC + P	Rende visibile il cursore
ESC + Q	Rende invisibile il cursore
ESC + Yyx	Posiziona il cursore in un punto definito dalle coordinate y' e x', dove y' rappresenta la riga (da 0 a 7) e x' la colonna (da 0 a 39). I valori di y e x da introdurre si ottengono aggiungendo 32 (codice ISO per lo Spazio) ai valori desiderati; il valore 0 corrisponde pertanto al codice ISO 32. Esempio: PRINT CHR$(27)+"Y"+" "+" " posiziona il cursore nell'angolo in alto a sinistra (0,0) Esempio: 10 A$="Olivetti M10" 20 PRINT CHR$(27)+CHR$(89)+CHR$(35)+CHR$(52)+A$ RUN <ENTER> visualizza le parole 'Olivetti M10' a partire dalla colonna 21 della riga 4; la linea 20 potrebbe anche essere scritta 20 PRINT CHR$(27)+"Y"+"£"+"4"+A$

E. PAROLE RISERVATE

L'elenco che segue contiene una serie di parole 'riservate', che hanno cioè un significato particolare in BASIC. Esse non devono pertanto essere usate come variabili, nomi di file, ecc.

§	EOF	LOAD	READ
ABS	EQV	LOC	REM
AND	ERL	LOF	RESTORE
ASC	ERR	LOG	RESUME
ATN	ERROR	LPOS	RETURN
BEEP	FILES	LPRINT	RIGHT$
CALL	FIX	MAX	RND
CDBL	FOR	MDM	RUN
CHR$	FRE	MENU	SAVE
CINT	GOSUB	MERGE	SCREEN
CLEAR	GOTO	MID$	SGN
CLOAD	HIMEM	MOD	SIN
CLOSE	IF	MOTOR	SOUND
CLS	IMP	NAME	SPACE$
COM	INKEY$	NEW	SQR
CONT	INP	NEXT	STEP
COS	INPUT	NOT	STOP
CSAVE	INSTR	OFF	STR$
CSNG	INT	ON	STRING$
CSRLIN	IPL	OPEN	TAB
DATA	KEY	OR	TAN
DATE$	KILL	OUT	THEN
DAY$	LCOPY	PEEK	TIME$
DEF	LEFT$	POKE	TO
DIM	LEN	POS	USING
DSKI$	LET	POWER	VAL
DSKO$	LFILES	PRESET	VARPTR
EDIT	LINE	PRINT	WIDTH
ELSE	LIST	PSET	XOR
END	LLIST		

F. IL MODEM COUPLER OLIVETTI MC10

Il modem/accoppiatore acustico (Modem Coupler) Olivetti MC10 è stato progettato per essere usato con il computer portatile Olivetti M10.

E' costituito da un modem per trasmissione dati e dai dispositivi per l'accoppiamento (meccanico ed acustico) al microtelefono di un apparecchio telefonico, privato o della rete pubblica. L'apparecchiatura consente di sfruttare pienamente l'opzione TELCOM di M10, per lo scambio di dati con un elaboratore remoto.

Le specifiche di MC10 sono conformi alle Raccomandazioni CCITT relative alla trasmissione dati su reti telefoniche commutate (Fascicolo VIII - Raccomandazione V.21).

MC10 è inoltre compatibile con sistemi dotati di interfaccia CCITT V.24/V.28 oppure EIA RS-232C. Opera in Full Duplex (duplice) a due canali: il canale 1 è associato al modo di funzionamento ORIGINATE/CAL, il canale 2 a quello ANSWER/ANS. Il primo modo viene utilizzato quando la chiamata ad un sistema ospite o ad una rete informativa è originata attraverso MC10, il secondo per rispondere alla chiamata di un altro computer.

MC10 richiede un'alimentazione in corrente continua a 12V, ottenuta mediante 8 pile a secco da 1.5V (da inserire nell'apposito scomparto) oppure con un alimentatore esterno collegato alla rete.

CARATTERISTICHE TECNICHE

Modalità di funzionamento:	full duplex, canale 1 e canale 2
Trasmissione dati:	binaria, seriale, asincrona
Velocità di modulazione:	300 baud
Frequenze in trasmissione:	canale 1 980 Hz (MARK) 1180 Hz (SPACE) canale 2 1650 Hz (MARK) 1850 Hz (SPACE)
Frequenze in ricezione:	canale 1 1650 Hz (MARK) 1850 Hz (SPACE) canale 2 980 Hz (MARK) 1180 Hz (SPACE)
Livello in trasmissione:	-18 dBm misurati all'uscita del telefono, verso la linea
Sensibilità in ricezione:	-43 dBm
Modulazione:	FSK (Frequency Shift Keying)
Tempo di ritardo nominale tra 'richiesta di invio' (C105) e 'pronto a trasmettere' (C106):	da OFF a ON, 30 ms da ON a OFF, 1 ms
Tempo di ritardo nominale del 'rivelatore di segnale di linea in ricezione' (C109):	da OFF a ON, 1000 ms da ON a OFF, 80 ms
Alimentazione:	8 pile da 1.5V AM, tipo IEC LR6, dimensione AA o UM/SUM-3 (in alternativa, può essere usato un alimentatore esterno)
Potenza dissipata:	0.5 W
Temperatura di esercizio:	da 0°C a 40°C
Umidità relativa:	sino al 90%
Interfaccia dati:	conforme alle Raccomandazioni CCITT V.24/V.28 o Standard EIA RS-232C
Circuiti di interfaccia:	accessibili mediante connettore maschio a 25 piedini (Fig. F-1)

PIN	CCITT NO.	COD.EIA	DESCRIZIONE
7	102	AB	Massa dei segnali
2	103	BA	Dati in trasmissione
3	104	BB	Dati in ricezione
4	105	CA	Richiesta di invio
5	106	CB	Pronto a trasmettere
6	107	CC	Data set (modem) pronto
20	108/2	CD	Terminale dati pronto
8	109	CF	Rivelatore di segnale di linea in ricezione
9	-	-	Controllo di tensione: +6V
10	-	-	Controllo di tensione: -6V

Fig. F-1 Circuiti di interfaccia

COMMUTATORI E INDICATORI LUMINOSI

Lo stato di MC10 è controllato e indicato rispettivamente da due commutatori e da un indicatore luminoso.

L'indicatore luminoso, posto sulla parte superiore, tra l'accoppiatore e il commutatore ON/OFF (Fig. F-2), fornisce le seguenti indicazioni:

- verde lampeggiante: indica che MC10 è acceso ma non in fase di comunicazione;

- verde continuo: indica che MC10 è acceso e in fase di comunicazione;

- rosso continuo: indica che la tensione delle pile è scesa al di sotto del valore minimo per un corretto funzionamento (le batterie devono essere sostituite al più presto).

Il commutatore posto accanto all'indicatore luminoso è l'interruttore di accensione ON/OFF.

Sul lato inferiore dell'unità si trova il commutatore ANS/CAL; questo commutatore è il selettore di canale:

- in posizione CAL seleziona il canale 1, secondo le Raccomandazioni CCITT o le specifiche EIA per ORIGINATE MODE. (questa posizione è utilizzata quando si effettua una chiamata verso sistema ospite);

- in posizione ANS seleziona il canale 2, secondo le Raccomandazioni CCITT o le specifiche EIA per ANSWER MODE. (questa posizione è utilizzata per predisporsi a ricevere una chiamata da un altro computer).

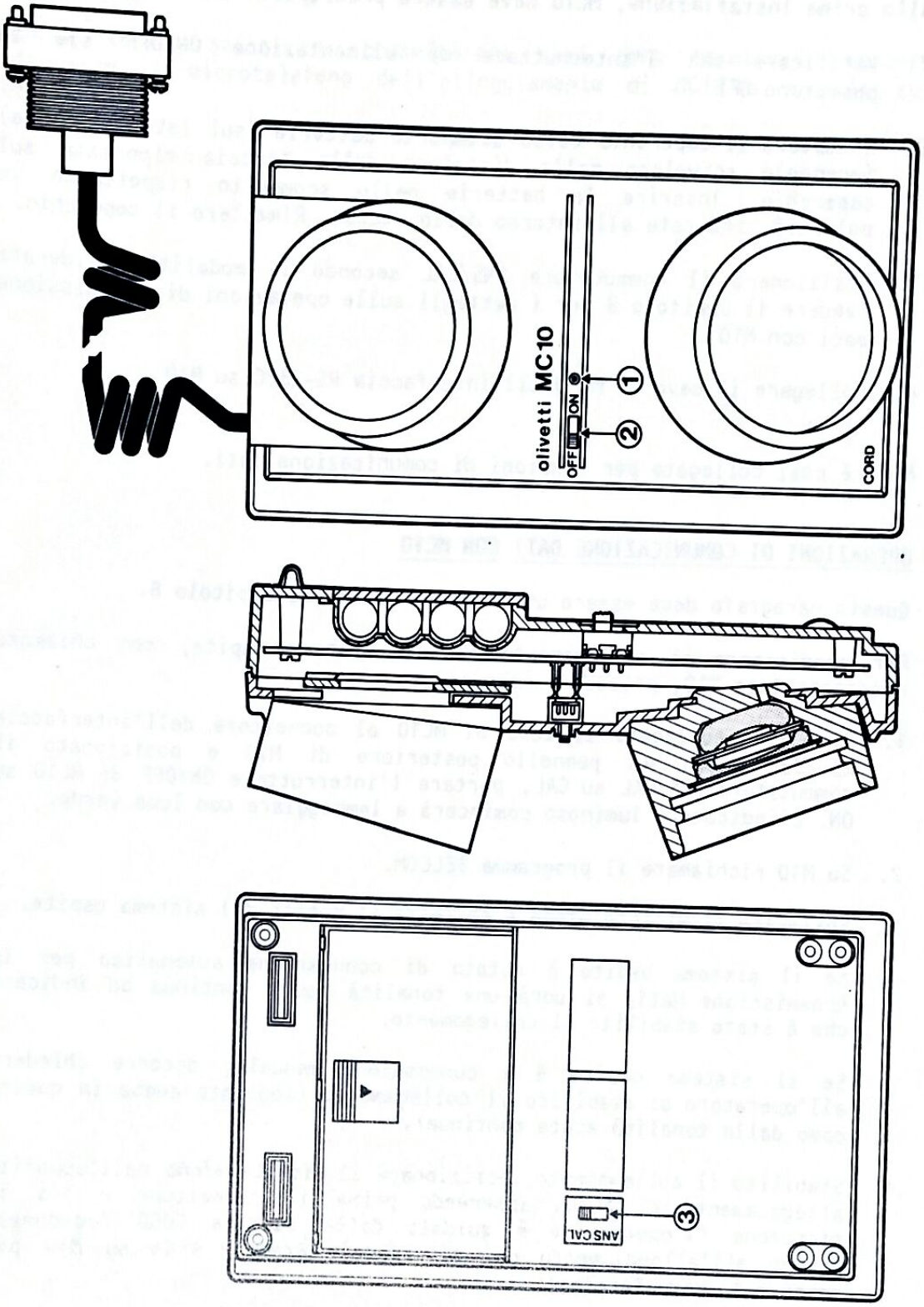

Fig. F-2 Il Modem Coupler Olivetti MC10

INSTALLAZIONE

Alla prima installazione, MC10 deve essere predisposto come segue:

1. Verificare che l'interruttore di alimentazione ON/OFF sia in posizione OFF.

2. Rimuovere il coperchio dello scomparto batterie (sul lato inferiore) facendolo scivolare nella direzione della freccia riportata sul coperchio. Inserire le batterie nello scomparto rispettando le polarità, indicate all'interno dello stesso. Rimettere il coperchio.

3. Posizionare il commutatore ANS/CAL secondo la modalità desiderata (vedere il Capitolo 8 per i dettagli sulle operazioni di trasmissione dati con M10).

4. Collegare il cavo di MC10 all'interfaccia RS-232C su M10.

MC10 è così collegato per funzioni di comunicazione dati.

OPERAZIONI DI COMUNICAZIONE DATI CON MC10

Questo paragrafo deve essere utilizzato assieme al Capitolo 8.

Per predisporre il collegamento con un sistema ospite, con chiamata originata lato M10, procedere come segue:

1. Dopo aver collegato il cavo di MC10 al connettore dell'interfaccia RS-232C posto sul pannello posteriore di M10 e posizionato il commutatore ANS/CAL su CAL, portare l'interruttore ON/OFF di MC10 su ON. L'indicatore luminoso comincerà a lampeggiare con luce verde.

2. Su M10 richiamare il programma TELCOM.

3. Sollevare il microtelefono e chiamare il numero del sistema ospite.

   Se il sistema ospite è dotato di connessione automatica per la trasmissione dati, si udrà una tonalità acuta continua ad indicare che è stato stabilito il collegamento.

   Se il sistema ospite è a connessione manuale, occorre chiedere all'operatore di stabilire il collegamento (indicato anche in questo caso dalla tonalità acuta continua).

4. Stabilito il collegamento, posizionare il microtelefono nell'apposito alloggiamento di MC10, inserendo prima il ricevitore e poi il microfono (l'operazione è guidata dalla scritta CORD (cordone), vicino all'alloggiamento del microfono). Premere <F4> su M10 per entrare in modo Terminal.

   Dopo un momento, l'indicatore luminoso emetterà una luce verde continua.

5. Il sistema è ora pronto per la trasmissione dei dati. Maggiori

dettagli sulla trasmissione dati con l'uso combinato di M10 e MC10 sono forniti nel Capitolo 8.

Quando le operazioni di trasmissione dati sono terminate, occorre rimuovere il microtelefono dall'alloggiamento di MC10 e riporlo sul telefono.

KTX 12 *** 07
Printed in Japan

olivetti